НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Інститут телекомунікаційних систем
Кафедра Телекомунікаційних систем

«До захисту допущено»
Завідувач кафедри
_____ Леонід УРИВСЬКИЙ
«___»_____20__ р.

Дипломна робота

на здобуття ступеня бакалавра

зі спеціальності 172 Телекомунікації та радіотехніка

на тему: «Розгортання SDN в корпоративних мережах на базі контролера APIC-EM фірми Cisco»

Виконав:
студент IV курсу, групи ТС-61
Пявчик Максим Олександрович _____

Керівник:
доцент кафедри ТС, к.т.н., доцент
Григоренко Олена Григорівна _____

Рецензент:
доцент кафедри ТК, к.т.н., доцент
_____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут телекомунікаційних систем
Кафедра Телекомунікаційних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 Телекомунікації та радіотехніка

Програма професійного спрямування (спеціалізація) – «Телекомунікаційні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____Леонід УРИВСЬКИЙ

«___»_____20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

Пявчику Максиму Олександровичу

1. Тема роботи «Розгортання SDN в корпоративних мережах на базі контролера APIC-EM фірми Cisco», керівник роботи Григоренко Олена Григорівна, доцент, к.т.н., затверджені наказом по університету від «30» березня 2020 р. №924-с

2. Термін подання студентом роботи 12 червня 2020

3. Вихідні дані до роботи: корпоративні мережі, контролер, обладнання фірми Cisco

4. Зміст роботи

1) Концепція та особливості побудови SDN;
2) Аналіз конфігурацій та особливостей використання контролера APIC-EM;
3) Практична реалізація взаємодії програм на базі контроллера APIC-EM;

4) Висновки.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

1) Презентація-захист на тему: «Розгортання SDN в корпоративних мережах на базі контролера APIC-EM фірми Cisco»

6. Дата видачі завдання 1 жовтня 2019 року

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз отриманого завдання	20.01.2020 – 01.02.2020	Виконав
2	Визначення мети дипломної роботи та розробка змісту	01.02.2020 – 10.02.2020	Виконав
3	Написання вступної частини дипломної роботи	10.02.2020 – 22.02.2020	Виконав
4	Написання першого розділу. Концепція та особливості побудови SDN	22.02.2020 – 07.03.2020	Виконав
5	Написання другого розділу. Аналіз конфігурацій та особливостей використання контролера APIC-EM компанії CISCO	07.03.2020 – 22.03.2020	виконав
6	Робота над третім розділом. Практична реалізація Взаємодії програм на базі контролера APIC-EM	22.03.2020 – 16.04.2020	Виконав
7	Написання висновків по трьом розділам	16.04.2020 – 04.05.2020	Виконав
8	Написання загального висновку	20.05.2020 – 23.05.2020	Виконав
9	Оформлення дипломної роботи	23.05.2020 – 30.05.2020	Виконав
10	Підготовка презентації до захисту	30.05.2020 – 07.06.2020	Виконав

Студент Максим ПЯВЧИК

Керівник роботи Олена ГРИГОРЕНКО

РЕФЕРАТ

Текстова частина дипломної роботи: 68 с., 37 рис., 1 табл., 13 джерел.

Метою роботи є дослідження SDN в корпоративних мережах на базі контролера APIC-EM фірми Cisco та створення програми віртуального трасування на обладнанні фірми Cisco.

В даній роботі розглянуто технології SDN та обладнання контролера APIC-EM.

У практичній частині показано створення програми, яка проводить трасування в корпоративній мережі і показує інформацію про пристрої, що були на її шляху .

КОРПОРАТИВНІ МЕРЕЖІ, SDN, МОВА ПРОГРАМУВАННЯ PYTHON, КОНТРОЛЕР APIC-EM КОМПАНІЇ CISCO SYSTEM

ABSTRACT

The purpose of the work is to study SDN in corporate networks based on the Cisco APIC-EM controller and to create a virtual tracing program on Cisco equipment.

In this paper, SDN technologies and APIC-EM controller equipment were considered.

The practical part shows the creation of a program that performs tracing in the corporate network and enters information about the devices that were in its path in the table.

CORPORATE NETWORKS, SDN, PYTHON PROGRAMMING LANGUAGE, APIC-EM CONTROLLER OF CISCO SYSTEM

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

APIC-EM — Application Policy Infrastructure Controller - Enterprise Module;
ACI — Application Centric Infrastructure;
ЦОД — центр обробки даних;
ООП — Об’єктно-орієнтоване програмування;
LAN — Local Area Network;
WAN — Wide Area Network;
SDN — Software-Defined Networking;
ACL — Access Control List;
VLAN — Virtual Local Area Network;
IoT — Internet of Things;
QoS — Quality of Service;
PoE — Power over Ethernet;
SFP — Small Form-factor Pluggable;
STP — Spanning Tree Protocol;
REST — Representational State Transfer;
CDP — Cisco Discovery Protocol;
API — application programming interface;

ВСТУП

Щосекунди кількість користувачів послуг Інтернет зростає. За даними агенства “We Are Social” і сервісу “Hootsuite” [1] на 2019 рік аудиторія мережі Інтернет зростає зі швидкістю 1 000 000 нових користувачів в день.

Це означає тільки одне, навантаження на системи та професіоналів, які обслуговують їх, зростають день за днем. Розроблюються нові підходи та системи, що дозволяють знизити навантаження з спеціалістів та розвантажити мережу та бізнес.

Метою роботи є дослідження принципів побудови корпоративних мереж з розгортанням SDN на базі програмованих контролерів, розгляд призначення, складу і особливостей побудови пристрою APIC-EM компанії Cisco Systems, його функціональних можливостей та переваг над іншими подібними пристроями від інших вендорів телекомунікаційного обладнання.

Завдання роботи — розглянути на практиці основні моменти в керуванні корпоративними мережами за допомогою контролера APIC-EM компанії Cisco Systems, описати основні конструктивні елементи контролера.

Об`єкт дослідження дипломної роботи — програмно визначаємі мережі (SDN).

Предмет дослідження дипломної роботи — розгортання SDN в корпоративних мережах на базі контролера обладнання компанії Cisco Systems.

Складність середовища стала перешкодою для своєчасної доставки сервісів і забезпечення якості обслуговування в корпоративних мережах. Необхідність швидко розгортати нові сервіси більше не залишає часу для ручних розрізнених процесів зміни мережі і впровадження додатків. Також є неприйнятною роз'єднаність управління мережею і налаштування політик. Замість цього необхідно інтегрувати мережеві ресурси і управляти ними динамічно за допомогою глобального уявлення мережі і платформи політик.

Щоб залишатися конкурентоспроможними в сучасному бізнес середовищі, що швидко змінюється, і не допускати виходу з під контролю витрат на експлуатацію мережі, потрібні повна прозорість роботи мережі і автоматичне налаштування і застосування політик.

Модуль Cisco APIC надає відкритий і програмований підхід до організації мережі за рахунок відкритих API-інтерфейсів для управління і забезпечення безпеки на основі політик. Це дозволяє автоматизувати процеси налаштування, які традиційно були трудомісткими і виконувалися вручну.

Контролери забезпечують узгоджену доставку мережевих сервісів і надають багато інформації та аналітичних даних про всі мережеві ресурси: LAN і WAN, дротові і бездротові підключення, фізичні і віртуальні інфраструктури. Такий прозорий контроль дозволяє оптимізувати обслуговування і підтримувати нові програми та бізнес-моделі. Контролер ліквідує розрив між відкритими, програмованими мережевими елементами і додатками, які взаємодіють з ними, автоматизуючи налаштування всієї інфраструктури.

1 КОНЦЕПЦІЯ ТА ОСОБЛИВОСТІ ПОБУДОВИ SDN

1.1 Концепція та переваги SDN

SDN (від англійського Software Defined Networking) — це одна з форм віртуалізації мережі. Головна особливість цих мереж полягає в тому, що їх рівень управління відділений від пристроїв передачі даних і реалізується програмно.[2]

Основні функції таких мережевих пристроїв, як маршрутизатор або комутатор, можна розділити на дві площини (рис.1.1):

Площина управління (Control Plane) — приймає рішення про переадресацію. Площина управління містить механізми переадресації маршруту рівня 2 та 3. Інформація, що надсилається до площини управління, обробляється процесором.

Площина даних (Data Plane) — також називається площиною переадресаці і використовується для перенесення потоків даних. Маршрутизатори та комутатори використовують інформацію від площини управління для передачі вхідного трафіку до відповідного інтерфейсу виходу.

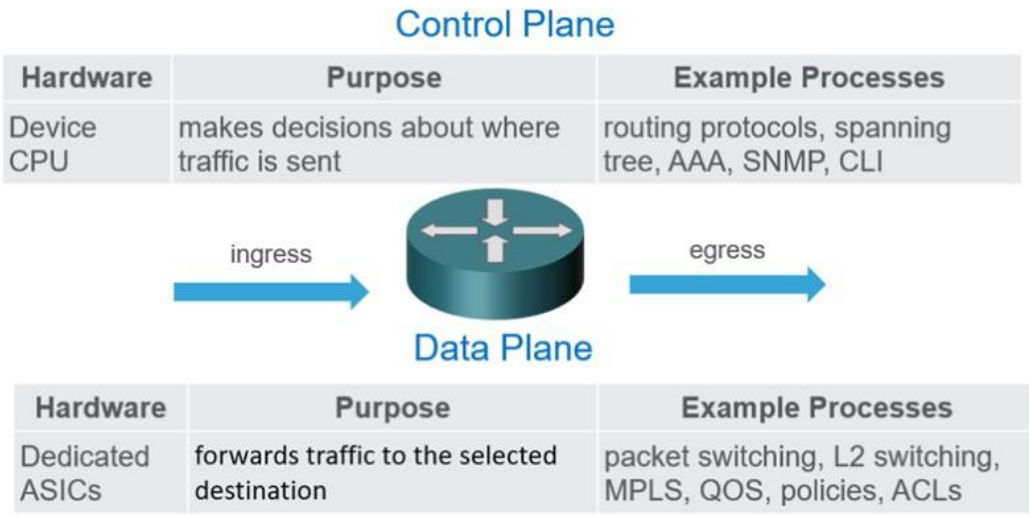


Рисунок 1.1 - Площина даних та площина управління

Розмежовуючи площину управління та площину даних, мережеві програмісти можуть централізувати інформацію, яку пристрої

використовують для прийняття рішень переадресації та виконання інших функцій.

Стрімке зростання обсягів трафіку і зміна його структури, необхідність підтримки зростаючої армії мобільних користувачів, формування високопродуктивних кластерів для обробки великої кількості інформації і спеціалізовані віртуальні середовища, що надають хмарні сервіси. Все це докорінно змінило вимоги до мереж. І нажаль, найчастіше саме цей фактор обмежує стрімкий розвиток обчислювальної інфраструктури.

Головною перешкодою залишаються класичні мережі, які характерезуються статикою, тому не відповідають динаміці, властивої сучасному бізнесу, на відміну від серверів — чим останні зобов'язані технології віртуалізації. Сьогодні додатки розподілені між безліччю віртуальних машин, які інтенсивно обмінюються даними. Для розвантаженя та балансування навантаження, сервери віртуальних машин часто міняють своє положення, що змінює точки «прив'язки» трафіку. Звичайні правила по адресації чи розподілу не відповідають сучасним викликам, тому вважаються застарівшими.

Наприклад, при запуску нової віртуальної машини, реконфігурування списків контролю доступу (ACL) на всіх мережевих пристроях у великій мережі може зайняти кілька днів. Причина — орієнтація наявних інструментів управління на роботу з окремими пристроями: в кращому випадку автоматизація призначення параметрів поширюється на групу пристроїв, в яку входять представники одного модельного ряду конкретного виробника. В результаті адміністраторам доводиться витрачати масу часу на те, щоб вручну перенлаштовувати правила обробки трафіку на кожному окремому пристрої. Такі ж проблеми виникають з переконфігурацією механізмів QoS при додаванні в мультисервісну мережу нового додатка, наприклад: відеозв'язку. Зміни в параметрах захисту вимагають достатньо велику кількість часу, що є обмежуючим фактором при швидкому реагуванні на можливі загрози.

Всі ці та вище згадані фактори унеможливлюють швидке та легке маштабування мереж. Додатково ускладнення виникають через лімітування по кількості логічних груп.. Наприклад, як відомо, стандартна технологія VLAN забезпечує підтримку лише 4096 віртуальних локальних мереж, а при розгортанні хмарних сервісів IaaS комерційним ЦОД вже сьогодні потрібно набагато більше число віртуальних мереж. Уявіть, що послуги IaaS надаються сотні підприємств, у кожного з яких по сотні VLAN, - вже в цьому випадку число логічних мереж становить десятки тисяч.

Архітектура традиційного мережевого обладнання (рис.1.3) робить цю «прив'язку» дуже міцною. SDN дає можливість істотно послабити, а то і повністю ліквідувати залежність замовників від технологій конкретного вендора.

В програмованих мережах важливі такі три речі:

- процеси передачі та управління даними розділені (рис.1.2);
- управління мережею централізовано за допомогою уніфікованих програмних засобів;
- обов'язкова віртуалізація фізичних мережевих ресурсів.

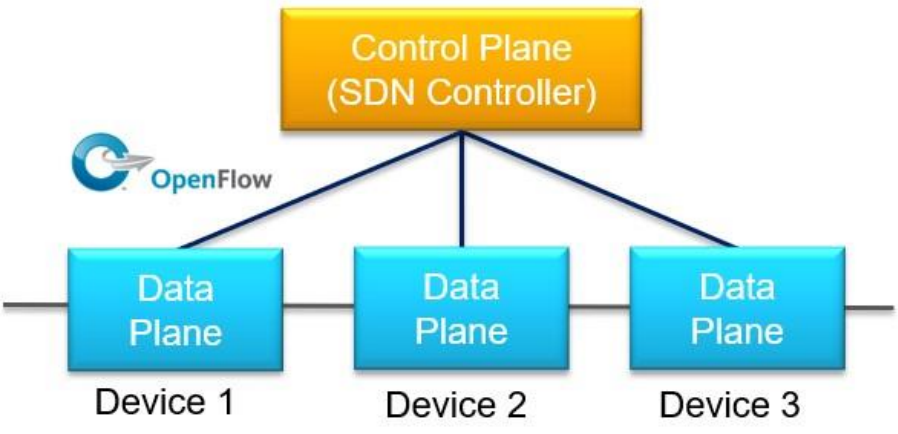


Рисунок 1.2 - SDN архітектура

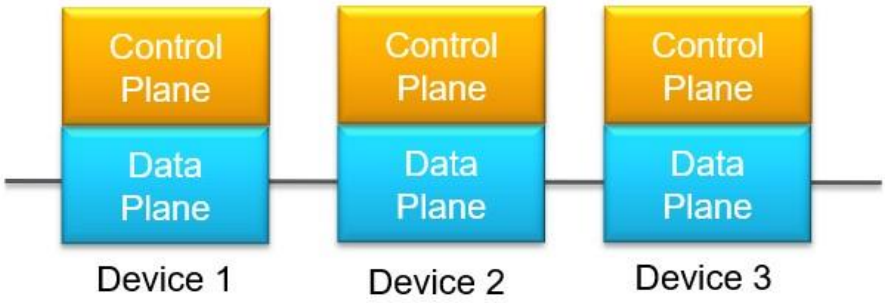


Рисунок 1.3 - Традиційна архітектура

В сучасних мережах технологіям SDN приділяється багато уваги, тому що області застосування є дуже перспективними і зараз набувають величезної популярності, найбільш вагомими є: [3]

- інфраструктурні хмарні сервіси, особливо у випадках необхідності швидкого виділення віртуальних ресурсів споживачам в автоматичному режимі;
- ЦОД;
- інтернет речей (IoT).

У випадку використання технологій SDN, вони мають над традиційними технологіями побудови мереж чотири основні переваги:

- По-перше, економія на капітальних витратах, як і при використанні хмарних технологій: закупівля серверів та мережевого обладнання, забезпечення безперервного процесу роботи і т.д.
- По-друге, скорочення часу на розгортання додаткових послуг і додатків в експлуатуємих ЦОД, а також на виділення додаткових віртуальних мережевих ресурсів.
- По-третє, зниження трудовитрат на супровід мережі (за рахунок централізації і автоматизації управління на програмному контролері) і на розгортання нових додатків.
- І підвищення ефективності використання ресурсів мережі за рахунок динамічного управління і практично миттєвої реакції на збільшення попиту на обчислювальні потужності.

1.2 Розгортання SDN в корпоративних мережах на базі контролерів

Будь-який сучасний бізнес користується послугами в сфері інформаційних технологій, а тому має повні підстави коректувати і пред'являти нові вимоги до гнучкості і масштабування телекомунікаційних мереж. Зважаючи на це, основними трендами в розвитку корпоративних мереж є:

• жваве зростання кількості споживаного трафіку і перелом його організації в бік передачі відео і уніфікованих комунікацій (UC-C);

• потреба підтримки нового погляду на влаштування робочих місць, так званих мобільних користувачів (BYOD);

• високопродуктивні кластери для обробки Великих Даних (BIG DATA);

• віртуалізація для надання хмарних сервісів (Cloud Bursting).

Традиційний вигляд мережі негативно впливає на розвиток обчислювальної інфраструктури. Стандартні методи для вирішення проблем, наприклад, на основі віртуалізації мереж (VLAN, VRF), ще не досягли рівня розвитку віртуалізації серверів і систем зберігання даних. Класично побудовані мережі дуже статичні і не задовольняють мтттєву динаміку розвитку сучасного бізнесу в сфері телекомунікацій. Стандартні мережі ще не спроможні маштабуватися з тією самою швидкістю, що і бізнес, який рухає прогрес уперед, а розподілене управління пристроями традиційних мереж занадто складне і не ефективне.

Світові потрібна нова технологія або бачення організації інформаційних мереж, що зможе вирішити проблеми, як були зазначені вище. Така технологія існує вже декілька років і носить назву — Software Defined Networking або скорочено SDN.

Перевагами в використанні Software Defined Networking (SDN) є:

• Розділення аовновадень між шарами площини управління і площини даних.;

• Спрощений та міжнародноприйнятий єдиний відкритий інтерфейс керування і передачі (OpenFlow);

• Управління мережею стане централізоване, а його центром буде виступати контролер (Контролер SDN);

• Більшість фізичних ресурсів в мережі стане віртуалізованими (що дасть можливість скинути баласт проблем з обслуговування даних компонентів);

• Можливості програмування, які обежені лише уявою спеціалістів, як обладнання (OpenFlow), так і додатків (API — Контролер SDN) сучасними і популярними мовами;

• Митєве реагувати на зміни, що відбуваються в мережі;

• Оптимізувати передачу трафіку (L2 / 3) через більшу кількість резервних шляхів;

• Легко і митєво налаштовувати мережі;

• Сутєво скорочується час на розгортання різних додатків;

• Спрощується керування мережевими пристроями;

• Простота керування всією мережею, а не окремими мережевими пристроями;

• Відкриті протоколи, засновані на стандартах, що дають змогу різним виробникам мережного обладнання взаємодіяти між собою, тим самим збільшуючи вибір для користувача і створюючи здорову конкуренцію між вендорами, прискорюючи інновації як в області програмного забезпечення, так і апаратних засобів;

• Контролер контролює всю мережу та трафік у ній[4].

Контролер SDN (рис.1.4) визначає потоки, які існують в площині даних. Кожен потік в мережі повинен спочатку бути дозволений контролером, який перевіряє, що потік не порушує мережеву політику. Якщо контролер дозволяє потік, він обчислює маршрут для потоку і додає запис для цього потоку в кожен комутатор, через який проходить потік. На відміну від складних функцій, що виконує контролер, комутатори просто

перенаправляють пакети відповідно до таблиць потоків, записи в яких можуть бути заповнені лише контролером. Зв'язок між контролером і комутаторами використовує стандартизований протокол і API. Найчастіше для цього використовується протокол OpenFlow.

Архітектура SDN відрізняється винятковою гнучкістю; можлива робота з різними типами комутаторів і на різних рівнях протоколу. Контролери і комутатори SDN можуть бути реалізовані для Ethernet-комутаторів (Layer 2), маршрутизаторів (Layer 3), транспорту (Layer 4) або маршрутизації на рівні додатків. SDN спирається на загальні функції, наявні в мережевих пристроях, які в основному пов'язані з пересилкою пакетів на основі визначення потоку.

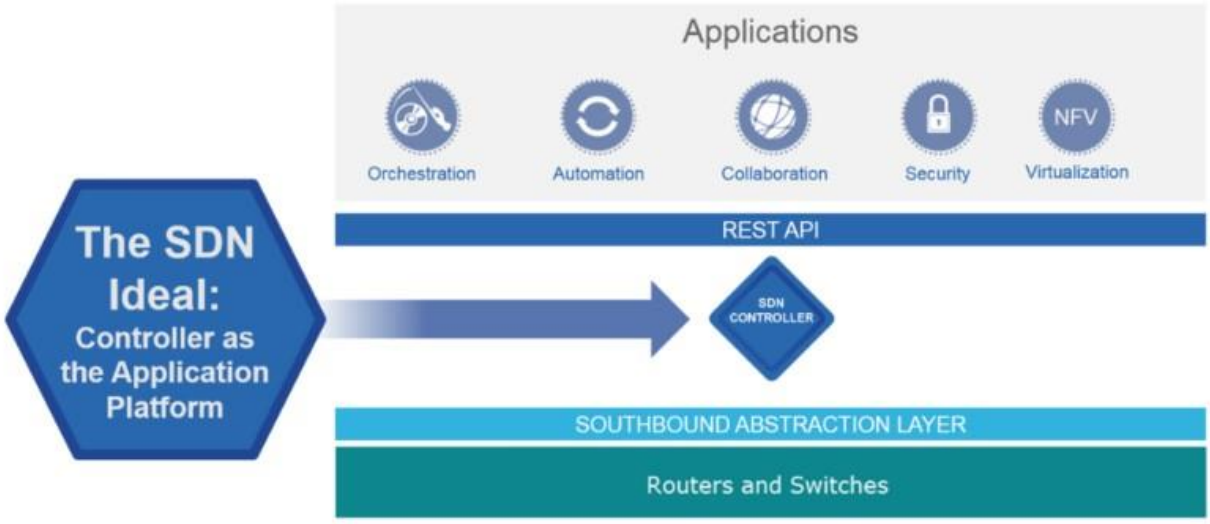


Рисунок 1.4 - Контролери в архітектурі

В архітектурі SDN-комутатор виконує наступні функції:

• Комутатор інкапсулює і перенаправляє перший пакет потоку в контролер SDN, щоб контролер міг вирішити, чи слід додати потік у таблицю потоку комутатора.

• Комутатор перенаправляє вхідні пакети з відповідного порту на основі таблиці потоків. Таблиця потоків може включати інформацію про пріоритет, продиктований контролером.

• Комутатор може відкидати пакети в певному потоці, тимчасово або постійно, як це продиктовано контролером. Викид пакетів може використовуватися для цілей безпеки, стримування атак типу відмова в обслуговуванні (DoS) або вимог до управління трафіком.

Таким чином, контролер SDN управляє станом пересилання комутаторів в SDN. Це управління здійснюється за допомогою API, що дозволяє контролеру задовольняти найрізноманітніші вимоги додатка без зміни будь-яких аспектів нижчого рівня мережі.

Завдяки розподілу площин управління і даних SDN дозволяє програмам працювати з одним абстрактним мережевим пристроєм, не піклуючись про деталі роботи пристрою. Мережеві додатки бачать в контролері один API. Таким чином, можна швидко створювати і розгортати нові додатки для організації потоку мережевого трафіку відповідно до конкретних корпоративних вимог продуктивності та /або безпеки.

1.3 Програмування в мережах SDN

1.3.1 Вибір мови програмування Python

На сьогоднішній день тенденція в телекомунікаціях йде на автоматизацію та спрощення управління обладнанням та процесами. Все більшу популярність набувають методи управління мережею типу контролерів та написання спеціальних скриптів на об'єктно-орієнтованих мовах типу: Python, Ruby, Java, C#, C++ та Swift.

Для легкого вибору мови програмування скористаємося трендами від Google. [5] Порівняємо три найбільш часто використовувані мови програмування, такі як: Python, Ruby, Java (рис.1.5 та 1.6). Лідером цієї трійки буде Python.[6]

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Python став свого роду феноменом, ця мова дуже стрімко набрала популярності серед програмістів, вчених і інженерів.

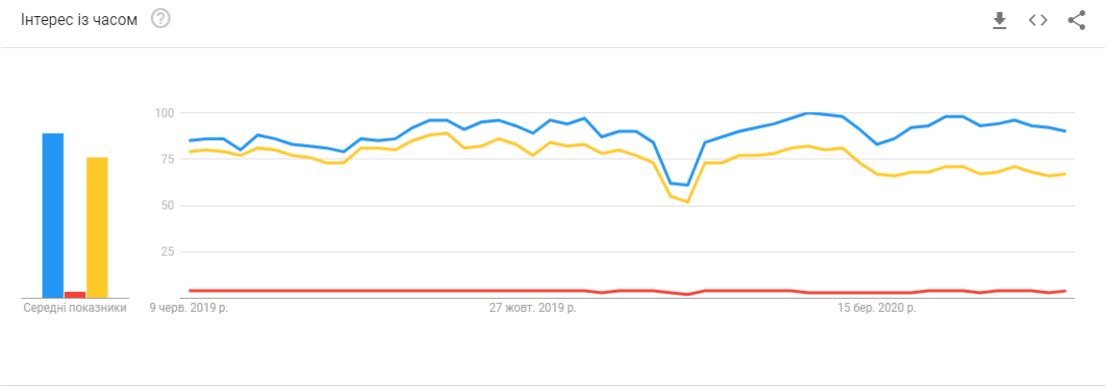


Рисунок 1.5 - Порівняння мов Python, Ruby і Java по популярності в світі

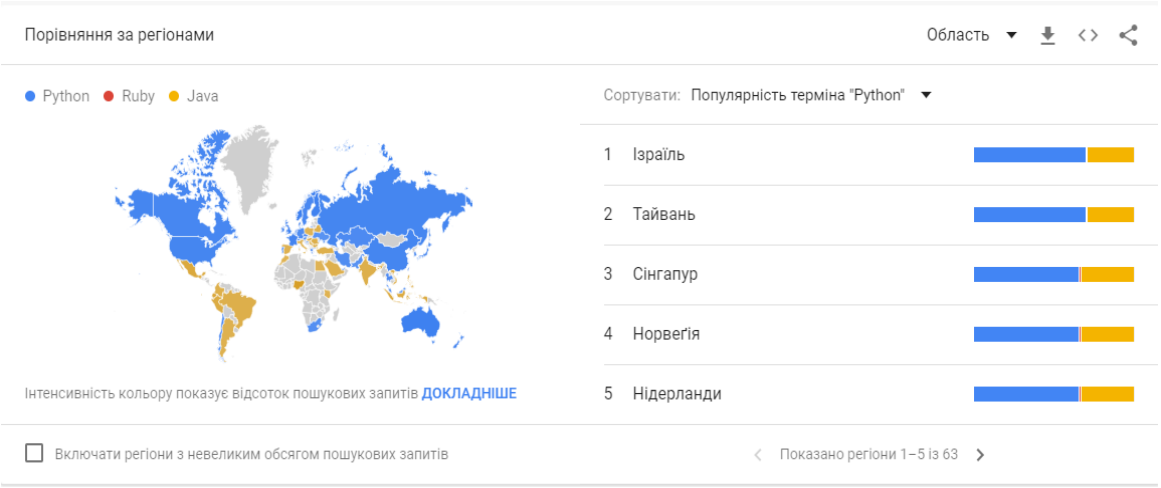


Рисунок 1.6 - Порівняння мов Python, Ruby і Java по регіонам

Вибір мови програмування Python – це простота розуміння і лаконічність. Саме таку філософію поставив перед собою розробник цієї мови, Гвідо ван Россум.

Головними перевагами Python серед інших мов можна виділити наступні:

- Швидкість розробки. Порівнюючи з іншими мовами, такими як C++, C# і Java в Python програмістам для написання однакових програм чи для вирішення математичних завдань потрібно витратити значно меньший час на написання коду, налагодження і супровід. Крім того, саме Python доволяє запустити програми зразу ж, не очікуючи компіляції, що в деяких випадках грає ключову роль у виборі мови написання;
- Перенесення програм. Для багатьох мов програмування перенесення програми з одної платформи на іншу стає справжньою Сізіфовою працею. Так як переважна частина програмістів пише свої програми на операційних системах типу Linux, а переважна частина користувачів цих програм використовує операційні системи на Windows, тому дуже важливим фактом є те, що для переносу спеціалісту потрібно лише скопіювати файли і запустити їх на потрібній операційній системі. Більш того, Python дає змогу створити графічні інтерфейси, програми доступу до баз даних і веб-додатків, що можна буде легко переносити. Крім того, в нових операційних системах Python вже предвстановлено з самого початку;
- Бібліотека підтримки. Python дуже популярний, тому спеціалістам буде дуже просто знайти потрібну бібліотеку в інтернеті на спеціальному сайті [7]. Також в самій мові присутня стандартна бібліотека, в якій зібрано безліч можливостей від пошуку по шаблону до спеціалізованих мережевих функцій. Серед списку сторонніх розробок можна вказати інструменти по створенню ігор, веб-сайтів, розробці математичних обчислень та багато іншого;
- Інтеграція компонентів. Сценарії написані на Python можуть з легкістю взаємодіяти з іншими частинами додатків завдяки своїм механізмам інтеграції. Програмний код на мові Python може визивати функції з бібліотек таких мов як: C++, C#, та сам визиватися з програм написаних на цих мовах;

- Легкість в засвоєні. Так як ця мова програмування з самого початку була створена для великого кола людей, вона була максимально спрощена для розуміння, в ній немає великих обсягів коду для вирішення необхідних завдань. Все просто і лаконічно. Головним девізом мови програмування на Python можна вказати таке «чим меньше коду, тим краще».

Останньою перевагою є якість програмного забезпечення. Однією з вагомих причин чому Python став насільки популяроною мовою є те, що він не потребує якихось фундаментальних знань, а може бути засвоєним будь-ким в дуже короткий проміжок часу. При написанні програм на мові програмування Python не потрібно постійно звертатися до інструкцій – це значуща перевага, завдяки якій на виході можна отримати професіонально написаний код, який може зрозуміти кожен.

Звичайно, Python має і недоліки. Головним недоліком спеціалісти вважають швидкість виконання програм, котра не завжди може бути порівняна зі швидкістю виконання таких компілюючих мов програмування як C++ або C#. Але легкість читання та написання коду є більш значущими. Також як було зазначено, Python – це жива мова програмування і вона постійно оновлюється, дописується, оптимізується та покращується і в деяких випадках, наприклад, коли обробляється файл чи конструюється графічний інтерфейс, то програма фактично виконує ці операції зі швидкістю, яку може показати мова програмування C, тому що такого роду задачі вирішуються компільованим з мови C програмним кодом, який лежить в основі інтерпретатора Python. Також швидкість виконання команд може бути зневільована потужністю сучасних комп’ютерів і їх процесорами.

Дати кількісну оцінку користувачів, які використовують мову програмування Python дуже важко, бо вона є безплатною і кожен може завантажити її собі з офіційного сайту, також як вже було зазначено у переважної більшості нових операційних систем мова програмування Python вже є предвстановленою. Так мова програмування Python є за замовчуванням

включена в дистрибутиви Linux, надається разом з комп'ютерами на базі Macintosh та деякими іншими програмними і апаратними продуктами.

Мова програмування Python була розроблена в 1990 році і за весь час свого існування включно до наших днів сколотила навколо себе цілу армію фанів, які дуже активно просувають її та вдосконалюють, крім того безліч великих компаній використовують Python для створення продуктів, які приносять шалений прибуток, до таких компаній можна віднести:

- Компанія Google дуже широко використовує різні мови програмування, в тому числі і мову програмування Python. Найяскравішим продуктом, який знають всі від цієї компанії є популярний відеохостинг, що надає послуги розміщення відеоматеріалів YouTube. Також популярний веб-фреймворк App Engine за свою основу взяв Python як прикладну мову програмування;
- Популярна програма BitTorrent для обміну файлами в парингових мережах (Peer-to-peer) реалізована на мові програмування Python;
- Такі гіганти, як Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm та IBM, використовують мову програмування Python для тестування апаратного забезпечення;
- Компанії як Industrial Light & Magic, Pixar, та інші часто використовують мову програмування Python в виробництві анімаційних фільмів;
- NASA, Los Alamos, Fermilab, JPL та інші використовують мову програмування Python для наукових досліджень;
- NASA також використовує Python для шифрування та аналізу даних;
- JPMorgan Chase, UBS, Getco и Citadel використовують мову програмування Python задля прогнозування фінансового ринку.

Таким чином, мову програмування Python можна пристосувати для вирішення як короткочасних задач та різних тактичних рішень, так і для більш стратегічних та довгострокових проектів. Як можна бачити з успішних

прикладів компаній та їхніх продуктів, Python гарно зарекомендував себе на обох теренах.

Python ідеально підходить для вирішення рутиних завдань. Мову програмування Python можна дійсно вважати багатофункціональною. Вона може бути використана, як для керування сторонніми програмними компонентами, так і для реалізації самостійних програм. Можна з повною впевненістю сказати, що Python являє собою багатофункціональне рішення для багатьох завдань і можливості мови майже не обмежені. На мові програмування Python можливо реалізувати все що завгодно, від тестування апаратних рішень, до керування безпілотними кораблями чи навіть супутниками. Найбільш вживаніші на даний час області застосування мови програмування Python:

- Ігри, зображення та штучний інтелект
- Програмування математичних чи наукових обчислень
- Додатки баз даних
- Системне програмування [8]
- Графічні інтерфейси
- Веб-сценарії

Узагальнюючи, багато з цих областей застосування Python - всього навсього різновиди однієї і тієї ж ролі під назвою «інтеграція компонентів». Використання Python в якості інтерфейсу до бібліотек компонентів, написаних мовою С, робить можливим створення сценаріїв на мові Python для вирішення завдань в самих різних прикладних областях. Як універсальну, багатоцільову мову програмування, що підтримує можливість інтеграції, Python може застосовуватися дуже широко.

Дивлячись на ці категорії, стає зрозуміло, що мова програмування Python – є дуже потужним інструментом. Сильні сторони Python:

1. Python об’єктно-орієнтована мова. Така об'єктна модель підтримує такі поняття, як поліморфізм, перевантаження операторів і множинне спадкування, однак, враховуючи простоту синтаксису і типізації Python,

Об’єктно-орієнтоване Програмування (ООП) не викликає складнощів в застосуванні.

2. Python безкоштовний. Python може використовуватися і розповсюджуватися абсолютно безкоштовно. Як і у випадку з іншими відкритими програмними продуктами, такими як Tcl, Perl, Linux і Apache, є можливість отримати в Інтернеті повні вихідні тексти реалізації Python. Немає ніяких обмежень на його копіювання, вбудову в системи або поширення в складі продуктів користувачів. Фактично користувач може навіть продавати вихідні тексти Python, якщо з'явиться таке бажання.

3. Python дуже простий в вивченні. Не потрібно витрачати купу часу на засвоєння специфічного синтаксису. Потрібно всього пару тижнів для того, щоб опанувати Python на базовому рівні. Це дуже гарний приклад того, чому саме Python вибирають більшість початківців і спеціалістів з суміжних спеціальностей з програмуванням, наприклад, мережеві інженери.

4. Python дуже зручний. Для того, щоб запустити програму потрібно лише написати в пошуку її ім’я і натиснути на іконку. Не потрібно виконувати ніяких компіляцій чи інших маніпуляцій з написаним кодом, як наприклад, в мовах програмування як C++ або ж C#. Інтерпретатор в Python спроєктований таким чином, що результат виконання програми отримується зразу ж після її запуску. В більшості випадків результат отримується миттєво, зразу ж після написання доповнень на клавіатурі.

5. Програми написані на мові програмування Python можуть бути з’єднані з іншими програмами, написаними на інших мовах. Як приклад можна навести прикладний інтерфейс C API в Python, який дозволяє програмам написаним на мові C визиватися та бути визваними з програм, написаних на мові Python.

1.4 Висновки до розділу 1

В першому розділі було наведено основні концепції та особливості побудови SDN, були перераховані переваги архітектури SDN над традиційною архітектурою. Зазначено, що в SDN процеси передачі та управління даними розділені; управління мережею централізовано за допомогою уніфікованих програмних засобів; обов'язкова віртуалізація фізичних мережевих ресурсів.

Також було рогляуто розгортання SDN в корпоративних мережах на базі контролерів, а саме контролерів від компанії Cisco Systems. З′ясовано, що контролер SDN управляє станом пересилання комутаторів в SDN. Це управління здійснюється за допомогою API, що дозволяє контролеру задовольняти найрізноманітніші вимоги додатка без зміни будь-яких аспектів нижчого рівня мережі. Також завдяки розподілу площин управління і даних SDN дозволяє програмам працювати з одним абстрактним мережевим пристроєм, не піклуючись про деталі роботи пристрою.

За допомогою Google trands була представлена найбільш популярна мова програмування для керування контролерами – Python.

Cisco APIC-EM — це перший комерційно доступний SDN контролер Cisco для корпоративної кампусової і розподіленої дротової і бездротової мережі. Це рішення органічно доповнює отримала світове визнання архітектуру Cisco ACI для центрів обробки даних (ЦОД). В рамках ACI функцію SDN-контролера виконує Cisco APIC. Таким чином з випуском Cisco APIC і потім Cisco APIC-EM, SDN-рішення Cisco повністю забезпечує автоматизацію IT процесів і послуг, що охоплюють всі домени корпоративної мережі.

2 АНАЛІЗ КОНФІГУРАЦІЙ ТА ОСОБЛИВОСТЕЙ ВИКОРИСТАННЯ КОНТРОЛЕРА APIC-EM КОМПАНІЇ CISCO

2.1 Архітектура побудови корпоративних мереж

Будь-яку організацію можна порівняти зі складним механізмом, в якому є велика кількість деталей, крім того вони взаємодіють між собою, допомагаючи виконувати роботу один одного, але кожна з них працює над однією задачею. Так само і в бізнесі, де існують багато відділів, що складаються з груп людей, які працюють над багатьма різними ідеями і проблемами, але все одно вони всі разом виконують одне головне завдання.

З розвитком організації та зростанням складності її побудови, дирекція починає замислюватися над впровадженням гнучкого та зручного контролю над роботою, що ведеться, та створенням максимально зручних умов для працівників для запобіганню втрати ресурсів та грошей. [9]

Для того щоб не витрачати зайвий час, гроші та ресурси, не переживати за конфіденційність інформації, що буде передаватися, використовують корпоративні мережі.

У основі будь-якої корпоративної мережі лежить апаратний шар, який включає комп'ютери різних класів. Набір комп'ютерів в мережі повинен відповідати набору різноманітних завдань, що вирішуються мережею.

Другий шар складає різноманітне мережеве устаткування, необхідне для створення локально-обчислювальних мереж, і комунікаційне устаткування для зв'язку з глобальними мережами. Комунікаційні пристрої грають не менш важливу роль, чим комп'ютери, які є основними елементами по обробці даних.

Третім шаром є операційні системи, які складають програмну основу мережі. При побудові мережевої структури важливо враховувати наскільки ефективно дана операційна система може взаємодіяти з іншими операційними системами мережі, наскільки вона здатна забезпечити безпеку і захист даних.

Самим верхнім шаром мережевих засобів є різні мережеві застосування, такі як мережеві бази даних, поштові системи, засоби архівації даних і ін. Важливо знати сумісність різних мережевих застосувань.

Основний дизайн архітектури корпоративної мережі має на меті впровадження такий цілей:

- Простота впровадження - розгортання рішення в найкоротші терміни;
- Гнучкість і можливість митєво масштабуватися — розподілена архітектура дає можливість вносити такі зміни, які необхідні саме в даний момент часу, з можливістю нарозення інфраструктури в майбутньому;
- Міцність та безпека — захист користувацького трафіку, допомагає виконувати стабільну роботу в мережі навіть під час серйозних кібератак;
- Простота управління — централізоване управління всією мережевою інфраструктурою;
- Готовність до впровадження нових технологічних рішень — впроваджена архітектура дозволяє легко адаптуватися до впровадження нових технологій і сервісів. (рис.2.1).

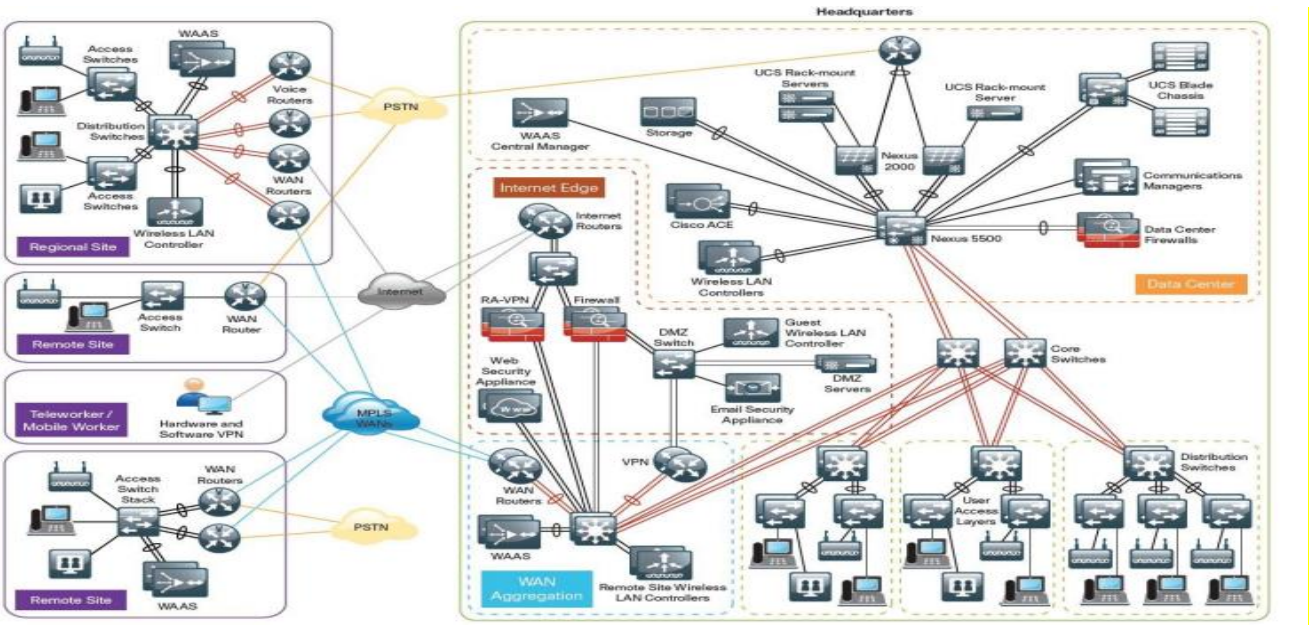


Рисунок 2.1 - Типова архітектура корпоративної мережі

Одним з основних критеріїв при побудові корпоративних мереж є модульність. Розбивши архітектуру мережі на модулі, можна сконцентруватися на функціоналі кожного з них окремо, що істотно спрощує дизайн, впровадження та управління. Створені модулі, як деталі конструктора, з яких можна зібрати мережу, відповідну вимогам. Ці ж деталі можна застосовувати повторно (реплікація), сильно скорочуючи час проектування. Принцип реплікації (повторення) елемента спрощує масштабованість мережі і прискорює її розгортання. На рисунку 2.2 показаний процес масштабування мережі.

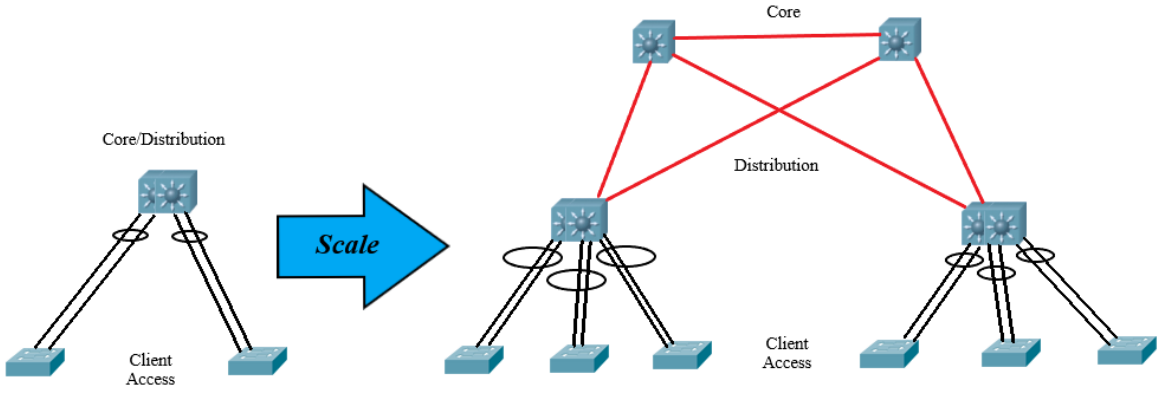


Рисунок 2.2 - Масштабування мережі

При розкладані топології на маленькі, прості для розуміння, частини дає змогу швидше та якісніше локаліщувати виникаючу проблему та митєво усунути її.

Ієрархічну модель можна порівняти з фундаментом на якому будується вся мережова інфраструктура, а саме: підключення абонентів, пристрїв різного характеру (телефонів, планшетів, портативних комп’ютерів), WAN маршрутизаторів, безпекових сервісів та обладнання.

Ієрархічна модель (Рис. 2.3) розділяє топологію мережі на три основні частини /рівня. Частини/рівні ієрархічної моделі:

- Рівень доступу (Access Layer) — на цьому рівні робота відбувається з кінцевими користувачами, їм надається послуги у вигляді підключення до мережі;
- Рівень розподілу (Distribution Layer) — цей рівень стоїть вище в ієрархії і виступає своєрідним концентратором, об’єднуючи рівні доступу в одну спільну групу;
- Рівень ядра або ще називають базовим рівнем (Core Layer) — найвищий рівень, який керує всією мережею, в ньому об’єднані рівень доступу і рівень розподілу.

Перераховані вище рівні несуть в собі різні функціональні можливості. Зважаючи на потреби, використовують кілька рівнів або усі одразу. До прикладу, якщо роглядати мережу, де кількість кінцевих користувачів меньше від десяти, то є сенс впроваджувати лише рівень доступу. Якщо роглядати мережу компанії чий розмір займає декілька поверхів або весь будинок, то кращим варіантом в цьому випадку буде впровадження двох рівнів доступу і розподілу. Якщо ж корпоративна мережа компанії розгорнута в декількох будівлях, то вкрай важливо використовувати усі три рівні: доступу, розподілу і базовий рівень.

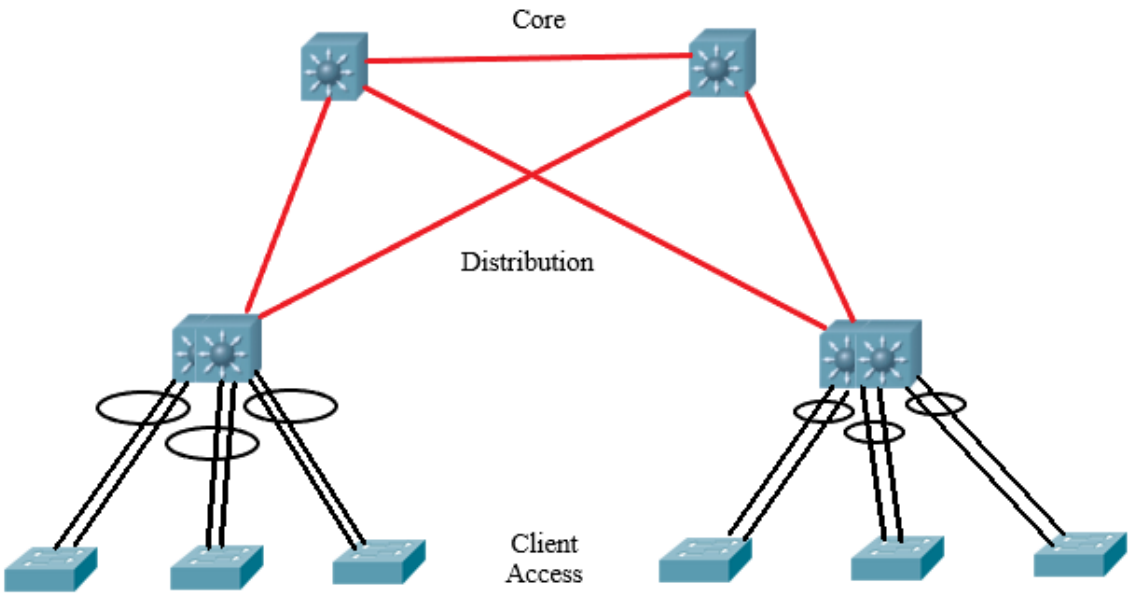


Рисунок 2.3 - Ієрархічна модель

Розглянемо кожен рівень детальніше.

2.1.1 Рівень доступу

Рівень доступу є однією з найголовніших частин всієї мережі, бо саме з нею найбільше стикаються кінцеві користувачі і якщо він буде побудований та організований без дотримання якості, то втрати від такого рішення буде підрахувти неможливо. (рис. 2.4) У більш ранній літературі даний рівень називається "Модуль доступу".

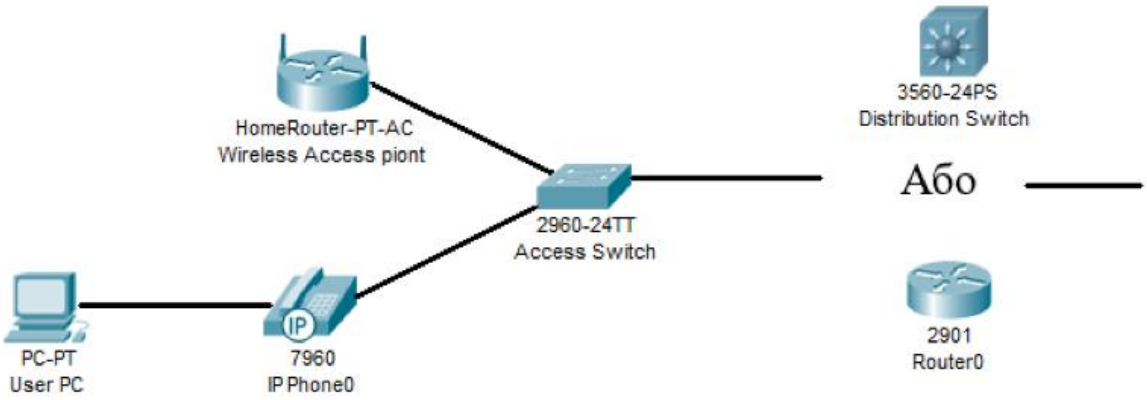


Рисунок 2.4 - Рівень доступу

За звичай до пристроїв, що працюють на рівні доступу включають різні комутатори другого рівня (L2), які працюють на канальному рівні моделі OSI. Такі пристрої заточені на впровадження початкових операцій в мережі, до прикладу створеня VLAN. Проте, як колись концентратори пішли в небуття історії, так само зараз вже і прості комутатири рівня (L2) рідко використовуються, частіше є можливість зустріти більш сучасні комутатори третього рівня (L3), бо цінова політика не є такою вже й сутєвою, а переваги в впровадженні даних апаратних рішень очевидні. Основним завданням пристроїв рівня доступу — це надавати високошвидкісне дротове (Gigabit Ethernet) і бездротове (802.11n) підключення до мережі.

Одним з основних завдань рівня доступу — це захист користувачів від різного роду кібератак, так як цей рівень, напряму вза’ємодіє з пристроями кінцеавих користувачів.

В рівень доступу входять наступні з перелічених технології, які захищають від даних стандартних мережевих атак:

- DHCP-snooping — атака при якій зловвмизник намагається підмінити DHCP-сервер і видати сабе замість нього;
- IP Source guard — атака при якій відбуваєится заміна IP адреси потрібного джерела на IP адресу зловмисника;
- Port security — відбуважться захист портів від підміни MAC адреси і від атак, спрямованих на переповнення таблиці комутації

Dynamic ARP inspection — захист від ARP spoofing-a, тобто від перехоплення мережевого трафіку між пристроями.

У разі якщо планується підключення до мережі таких пристроїв, як IP-телефони, IP-відеокамери або бездротові точки доступу, буде розумним використовувати комутатори з підтримкою технології PoE (Power over Ethernet). Це істотно спростить і здешевить впровадження вищевказаних пристроїв (виключається необхідність в додатковому живленні від електромережі).

Найбільш економічним рішенням є комутатори Catalyst серії 2960. Рішення на основі цих комутаторів надає найнижчу вартість за порт (підключеного користувача, сервера або будь-якого іншого пристрою), при цьому забезпечує весь необхідний функціонал для рівня доступу (сегментування мережі, QoS, PoE, і т.д.). Використання комутаторів рівня доступу дозволяє істотно знизити витрати на підключення користувачів і серверів. На даний момент в лінійці з'явилася нова, більш продуктивна і сучасна модель Cisco Catalyst 2960-X, вартість якої порівняна з вартістю попередньої моделі. При проектуванні мереж буде доречним використання нових комутаторів. Комутатори серії 3560, 3750, 4500 і 4507 застосовуються набагато рідше і тільки в тому випадку, коли покупка окремого комутатора для рівня доступу є недоцільною (мала кількість користувачів). Дані комутатори більше підходять для рівня розподілу.

У разі установки декількох комутаторів рівня доступу, розташованих в безпосередній близькості (в одній серверній шафі) рекомендується використовувати технологію стекування (рисунок 2.5).

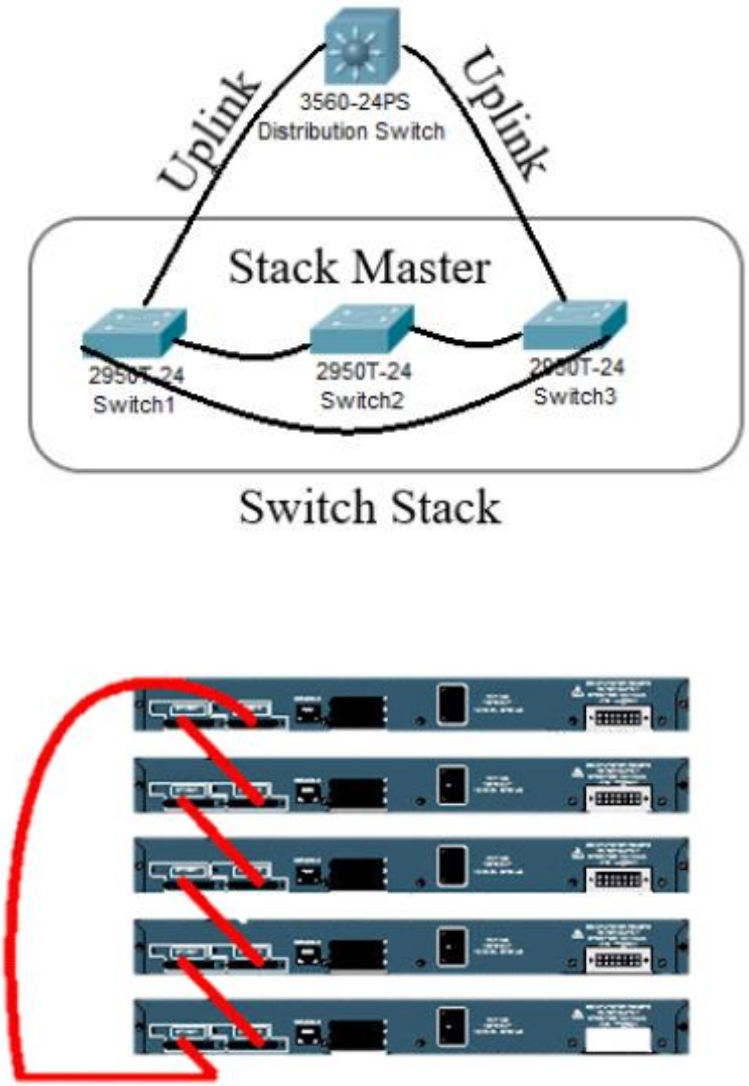


Рисунок 2.5 - Стек комутаторів рівня доступу

Дана технологія дозволяє об'єднувати обладнання в єдине ціле. Стек з трьох 24-х портових комутаторів використовується як один пристрій, що має 72 порти. Це істотно полегшує управління і конфігурацію, і також реалізує додаткову відмовостійкість. Однак слід зазначити, що дане рішення буде істотно дорожче, тому що вимагає придбання додаткових модулів стекування для комутаторів серії 2960-S і 2960-X.

Кожен комутатор рівня доступу повинен підключатися до комутаторів рівня розподілу по агрегованого каналу:

- Комутатори рівня доступу повинні розташовуватися не більше ніж в 90 метрах від користувачів (комутаційний шафа або серверна кімната) для їх підключення по крученій парі;
- Якщо пристрої рівня доступу знаходяться на відстані більш ніж 100 м від комутаторів рівня розподілу, то використовується оптоволоконное з'єднання. Це варто враховувати при проектуванні і закладати комутатори з підтримкою оптоволоконних підключень (технологія SFP, SFP +).

Варіант побудований не на пристроях Cisco.

Пристрої рівня доступу є найдешевшими в мережевій інфраструктурі, однак, їх може бути велика кількість, що веде до великих витрат. Вартість сучасного 24-х портового комутатора компанії Cisco (Catalyst 2960-X 24 GigE4 x 1G SFP LAN Base) становить близько 2400 $. При виборі інших моделей варто чітко розуміти, який функціонал буде потрібний від пристроїв рівня доступу. Комутатори другого рівня компаній D-link, Zyxel схожої конфігурації будуть коштувати дешевше в 2-3 рази. Такі комутатори підійдуть для підключення серверів. Для підключення користувачів можна використовувати дешевші рішення вище згаданих компаній, але тільки в тому випадку, якщо вимоги до безпеки не надто високі. Наприклад, комутатори D-link і Zyxel дуже поширені серед провайдерів інтернет, зважаючи на свою дешевизну і достатнього для їх завдань функціоналу.

2.1.2 Рівень розподілу

Рівень розподілу забезпечую доступ і правильну роботу сервісів. Основною задачею цього рівня є з’єднання в один моноліт декількох підпорядкованих йому рівнів доступу. Ця процедура забезпечує спрщення створення з’єднань. За звичай, саме до комутаторів цбого рівня підключають

найважливіші сервіси,такі як: модуль мережі Internet, модуль WAN мережі, та інші. (Рисунок 2.6.)

До пристроїв рівня розподілу, як правило, вносять (L3) комутатори. Ці пристрої мають більше можливостей, ніж їх молодші брати (L2) комутатори, наприклад маршрутизацію пакетів, а також на них можливо впровадити реалізацію систем безпеки і мережових політики для запобіганю порушень правил обміну пакетами вмережі.

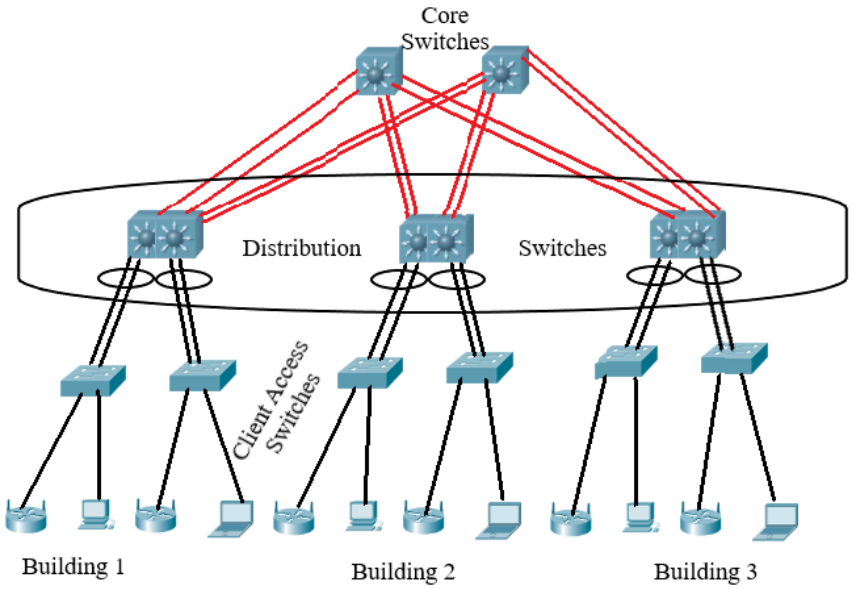


Рисунок 2.6 - Рівень розподілу

Обладнання, яке використовують для забезпечення прцездатності рівня розподілу, на основі обладнання компанії Cisco Systems:

- Cisco Catalyst 6500 E-Series 6-Slot Chassis;
- Cisco Catalyst 6500 VSS Supervisor 2T with 2 ports 10GbE and PFC4;
- Cisco Catalyst 6500 16-port 10GbE Fiber Module w/DFC4;
- Cisco Catalyst 6500 24-port GbE SFP Fiber Module w/DFC4;
- Cisco Catalyst 6500 4-port 40GbE/16-port 10GbE Fiber Module w/DFC4;
- Cisco Catalyst 6500 4-port 10GbE SFP+ adapter for WX-X6904-40G module;

- Cisco Catalyst 4507R+E 7-slot Chassis with 48Gbps per slot;
- Cisco Catalyst 4500 E-Series Supervisor Engine 7-E, 848Gbps;
- Cisco Catalyst 4500 E-Series 24-port GbE SFP Fiber Module;
- Cisco Catalyst 4500 E-Series 12-port 10GbE SFP+ Fiber Module;
- Cisco Catalyst 3750-X Series Stackable 12 GbE SFP ports;
- Cisco Catalyst 3750-X Series Two 10GbE SFP+ and Two GbE SFP ports network module;
- Cisco Catalyst 3750-X Series Four GbE SFP ports network module;

Також використовують наступні пристрої (але слід враховувати, що це Stand-Alone комутатори, тобто вони не здатні використовувати технологію стекування (на відміну від 3750-X), а це означає, що високопродуктивну і відмовостійку конфігурацію не можливо реалізувати на даних моделях комутаторів):

- Cisco Catalyst 3560-X Series Standalone 48 Ethernet 10/100/1000
- Cisco Catalyst 3560-X Series Standalone 24 Ethernet 10/100/1000

При проектуванні слід враховувати наступні загрози безпеці на рівні розподілу:

- Контроль доступу — атаки які здійснюються на корпоративні ресурси. Запобігаються простими обмеженнями політик безпеки так звані списки доступу;
- Захист від IP spoofing-у.

Для рівня розподілу захист від загроз не є основим завданням . Головне, що вимагається і що є найважливішим в цьому рівне це коректне і найбільш ефективне об’єднання рівнів доступу.

Рівень розподілу є дуже важливою ланкою в роботі всієї мережевої інфраструктури і вимагає високопродуктивного, відмовостійкого виконання.

Модель Cisco SBA LAN передбачає використання технології стекування і агрегованих з'єднань між мережевими пристроями, в той час як традиційна модель використовує принцип надмірності (redundant).

Нова модель SBA використовує агреговані канали між пристроями рівня доступу і рівня розподілу (з використанням таких протоколів як EtherChannel), одночасно забезпечуючи відмовостійкість і більш високу продуктивність. Агрегований канал є об'єднанням 2-х, 3-х або більше фізичних (проводових) з'єднань в одне логічне. При цьому всі з'єднання передають інформацію, що істотно збільшує пропускну здатність каналу (Рис. 2.7). У разі відмови одного із з'єднань, що входить в агрегований канал, інформація каналу, що відмовив, передається по іншим справним з'єднанням без будь-яких перерв у роботі мережі. Це вигідно на відміну від традиційної надлишкової моделі, в якій блокуються додаткові з'єднання (протокол STP, RSTP) для запобігання петель (Рис. 2.8). Таким чином, при використанні традиційної моделі продуктивність не збільшується, реалізується тільки відмовостійкість.

Комутатори рівня розподілу об'єднуються в стек (з використанням таких технологій як StackWise Plus). Агрегований канал утворюється при об'єднанні портів різних комутаторів стека (Рис. 2.9). Іншими словами, логічний інтерфейс утворюється об'єднанням двох (або більше) портів, при цьому один порт належить першому комутатору стека, а другий порт - другому. Обидва порти беруть участь в передачі трафіку. Таким чином виявляються задіяними всі пристрої, забезпечуючи високу продуктивність і відмовостійкість.

У традиційній надлишковій (Redundant) моделі мережевий трафік передає тільки один пристрій. Другий пристрій стає активним лише при відмові першого, або при відмові одного з активних з'єднань (спрацює технологія STP).

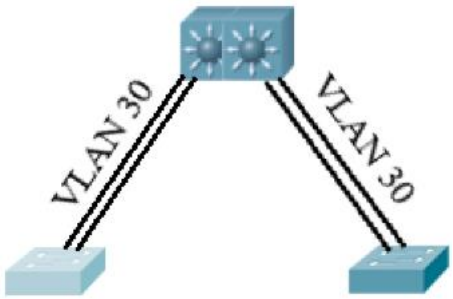


Рисунок 2.7 - Нова модель SBA LAN

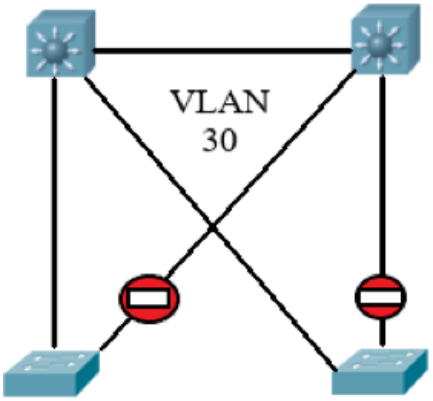


Рисунок 2.8 - Традиційна надлишкова модель.

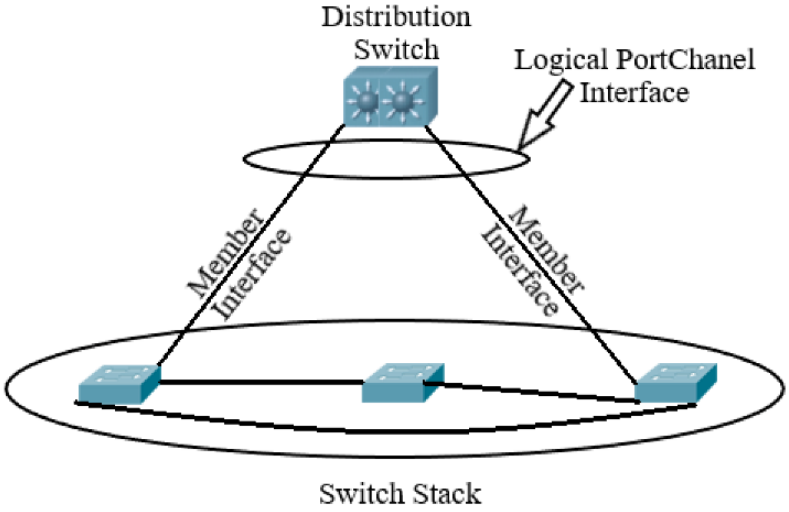


Рисунок 2.9 - Об’єднання портів стека комутаторів в один Port Chanel

Альтернатива побудована не на базі обладнання від компанії Cisco. Для зниження витрат і загального числа встановлюваних пристроїв можна

об'єднати рівень розподілу з рівнем ядра, якщо це дозволяють розміри мережі і вимоги до пропускної здатності. Це досить часта практика. Рівень розподілувиступає в якості рівня ядра і називається Collapsed core (рис.2.10).

В якості альтернативного обладнання можна вибрати рішення компанії Juniper або HP. Дані компанії є основними конкурентами компанії Cisco в корпоративному сегменті. Комутатори Juniper і HP трохи дешевше, однак якщо в мережевій інфраструктурі переважають комутатори (а так само міжмережеві екрани, IPS) компанії Cisco, то не варто використовувати обладнання різних вендорів заради невеликої економії. Набагато простіше управляти мережами, побудованими на обладнанні одного вендора (особливо, якщо це стосується обладнання компанії Cisco). Так само варто врахувати важливість підтримки технології стекування. Одним з найдешевших рішень є комутатори компанії D-Link. Наприклад, модель DGS-3120-24PC / B1ARI - L3 комутатор, що підтримує технологію стекування.

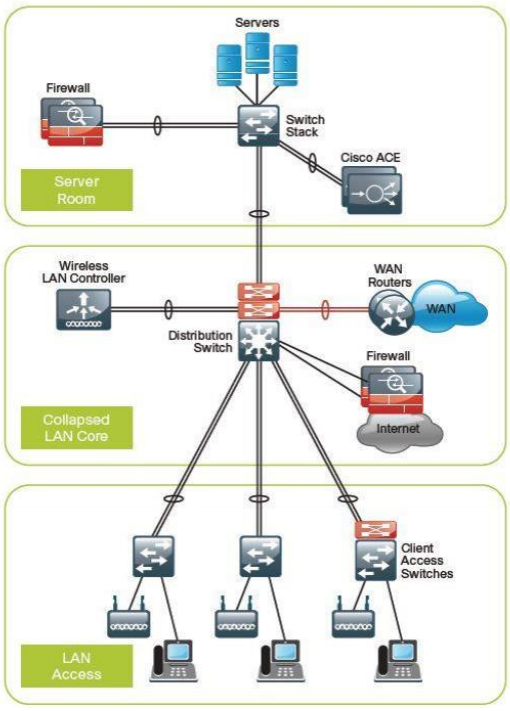


Рисунок 2.10 - Рівень розподілення об’єднаний з рівнем ядра (Collapsed core)

2.1.3 Рівень ядра

В нормах побудови корпоративних мереж, якщо мережа охоплює дві або більше будівель, тоді потрібно використання Рівня Ядра. Основною задачею рівня ядра є об’єднання всіх рівнів, тобто рівнів доступу і рівнів розподалення.Такий крок дозволяє мінімізувати кількість використаних з’єднань та зробити розробку мережі більш економною. На рисунках 2.11 і 2.12 представлені два різні уявлення про побудову мережі, без і з рівнем ядра відповідно. Зрозу ж кидається в очі значуща різниця в кількості з’єднань, що використовуються в побудові мережі без рівня ядра.(Рис.2.13.)

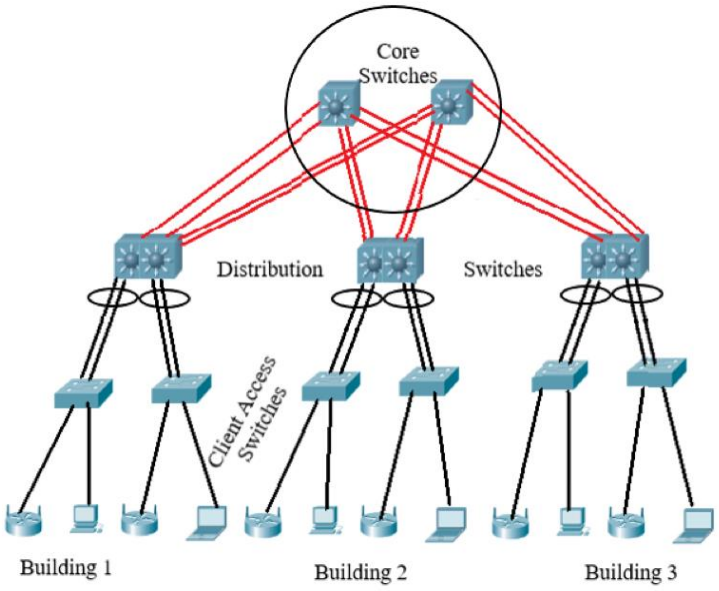


Рисунок 2.11 - Рівень ядра(Core Layer)

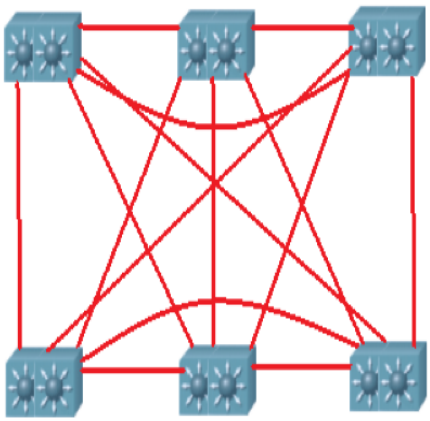


Рисунок 2.12 - Дизайн мережі без рівня ядра

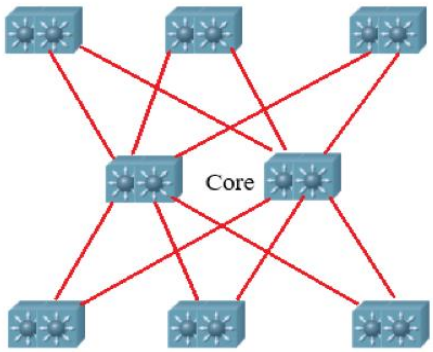


Рисунок 2.13 - Дизайн мережі з рівнем ядра

Комутатори, що знаходяться на рівні ядра не виконують якихось складних дій. Головне завдання комутаторів рівня ядра — це маршрутизація трафіку між частинами мережі. За звичай рівень ядра будується на двух комутаторах (L3)(резерв і основний).

Обладнання, яке частіше за все використовують в якості рівня ядра на базі обладнання вендера Cisco Systems це:

- Cisco Catalyst 6500 E-Series 6-Slot Chassis
- Cisco Catalyst 6500 VSS Supervisor 2T with 2 ports 10GbE and PFC4
- Cisco Catalyst 6500 24-port GbE SFP Fiber Module w/DFC4
- Cisco Catalyst 6500 4-port 40GbE/16-port 10GbE Fiber Module w/DFC4
- Cisco Catalyst 6500 8-port 10GbE Fiber Module w/ DFC4

Як вже було зазначено вище, осовною ідеєю рівня ядра є маршрутизація, а це означає, що забезпечення безпеки не є основним напрямком цього рівня. При завантаженні рівня ядра завданнями, крім маршрутизації, втрачається сама ідея швидкої і продуктивної мережі, тому такі кроки робити не рекомендується.

Об’єднання кількох будівель в одну корпоративну мережу не можливе без використання контрольованої зони. Контрольована зона — це власний канал передачі даних (оптичний, мідний і т.д.). Тобто якщо між двох будинків з'єднання здійснюється по спеціальному виділеному каналу, то в

цьому випадку можна організовувати рівень ядра. Але коли будівлі з'єднані через Інтернет канал, тоді в такому випадку варто використовувати спеціальний модуль — або так званий модуль Інтернет (Internet Edge). До рівня ядра підключаються всі частини мережі (комутатори рівня розподілу). В загальному вигляді схема підключення представлена на рисунку 2.14.

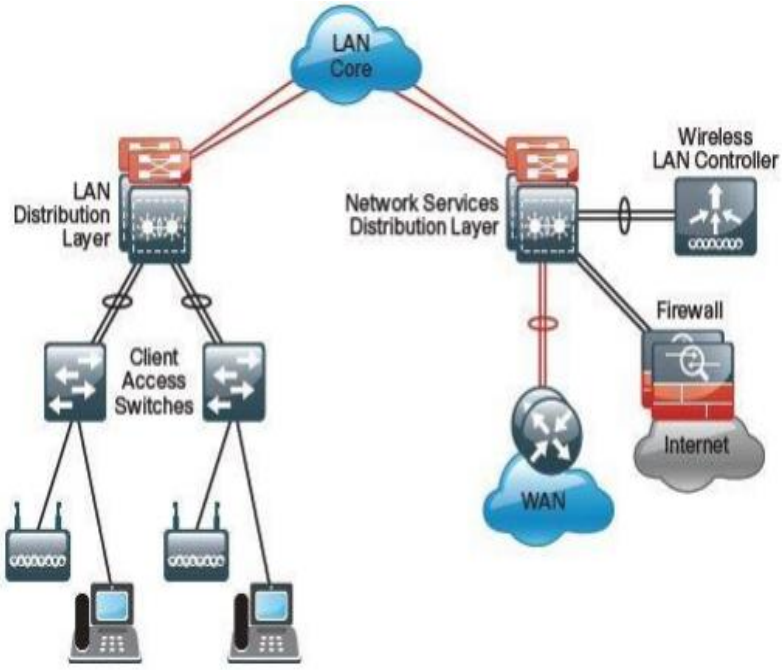


Рисунок 2.14 - Підключення модулів мережі до рівня ядра

Так, як комутатори рівня ядра підключаються насамперед до швидкісних інтерфейсі, вони повині володіти високою пропускною спроможністю серед всіх комутаторів корпоративної мережі (від 40 Гбіт / с). Комутатори рівня розподілу, повинні підключатися до обох комутаторів (резервного і основного) рівня ядра, тим самим забезпечуючи відмовостійкість. Підключення здійснюється з використанням технологій EtherChannel, що дозволяє балансувати потік трафіку. На рисунку 2.15 представлений приклад використання рівня Ядра.

Комутатори рівня Ядра є найдорожчими пристроями в ієрархічній моделі мережі (якщо розглядати тільки комутатори і не брати в розрахунок пристрою безпеки). Далеко не кожна організація може собі дозволити дані пристрої. Однак при необхідності використання рівня Ядра, в першу чергу потрібно визначитися з пропускною спроможністю, яка вимагається від

обладнання. Можливо, що для цілей організації підійдуть пристрої з більш дешевого сегмента (наприклад комутатори рівня розподілу). Так само необхідно розуміти, що одним з найважливіших параметрів рівня Ядра є відмовостійкість, тому що від пристрою даного рівня залежить робота величезної мережі (в маленьких мережах рівень ядра зазвичай відсутній або ж інтегрований з рівнем розподілу). Тому при виборі обладнання варто звертати увагу на технології організації відмовостійкості, резервування живлення.

В якості альтернативного обладнання, тобто не від вендора Cisco, можна вибрати наступні рішення. Лідерами серед комутаторів рівня ядра є компанії: Cisco, Juniper, HP, Brocade, Extreme Networks. Однак є й дешевші рішення рівня ядра від компанії D-Link.

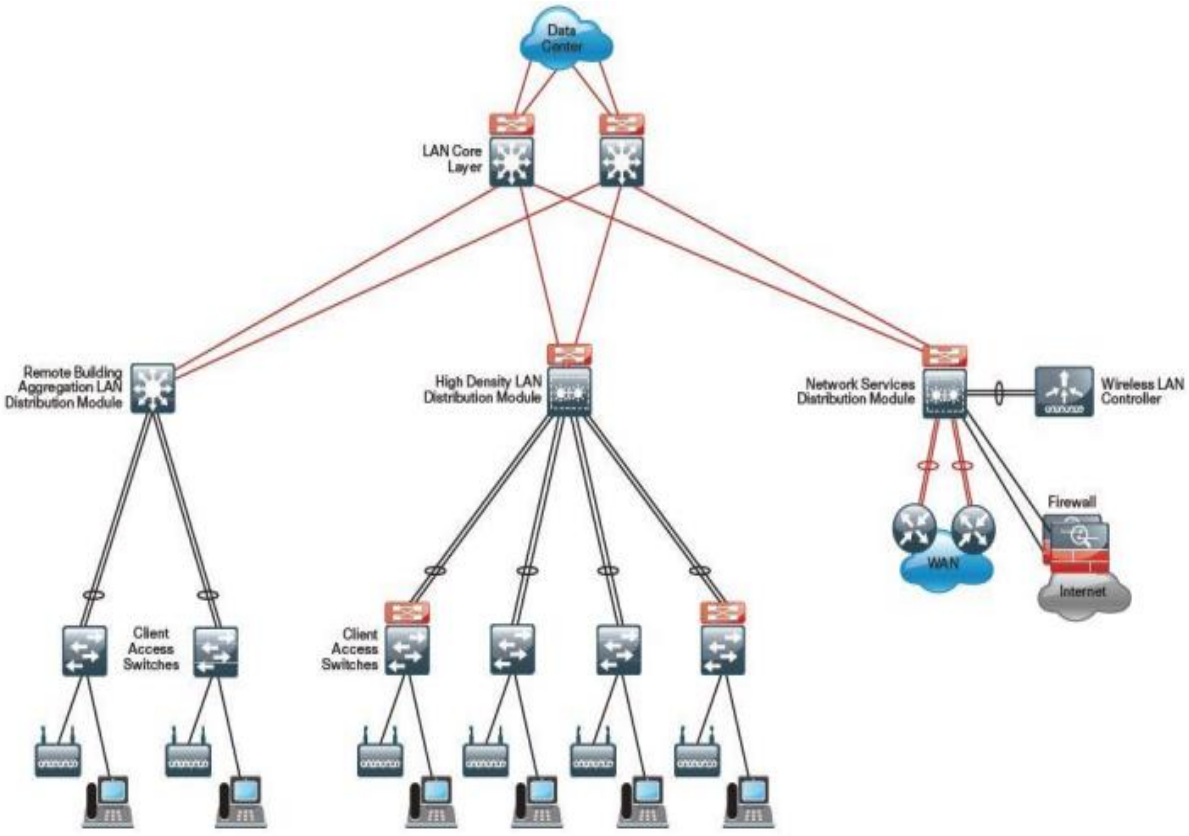


Рисунок 2.15 - Приклад використовування рівня ядра (LAN Core Layer)

2.2 Призначення, побудова та основні функції контролера APIC-EM компанії Cisco.

Центром будь-якого рішення в архутектурі SDN, безумовно, є контролер. Контролер використовує особливі інтерфейси для керування мережею, основний з них — OpenFlow. Компанія Cisco Systems розробила свій підхід до поділу мережі замовника на певні, окремі логічні домени (ЦОД, WAN, кампус, і т.д.) і впроваджує для кожного домена свій спеціалізований SDN-контролер, який максимально буде задовольняти потреби клієнта. Для забезпечення найкращого сервісу, що вимагає взаємодії кількох інфраструктурних доменів, Cisco має своє унікальне рішення по організації (Network Service Orchestrator, NSO). Для контроля за корпоративними мережами Cisco Systems пропонує впровадження контролера APIC-EM (Application Policy Infrastructure Controller — Enterprise Module), він був створений для керування корпоративними і розподіленими (WAN) мережами. Він є продовженням еволюції ланки контролерів типу APIC. (Application Policy Infrastructure Controller), що були створені для керування центрами обробки даних.

Контролер APIC-EM реалізує функціональність управління мережевими елементами, залишаючи всі інші завдання зовнішнім системам управління і стороннім додаткам, які взаємодіють з APIC-EM через «північний» програмний інтерфейс REST.

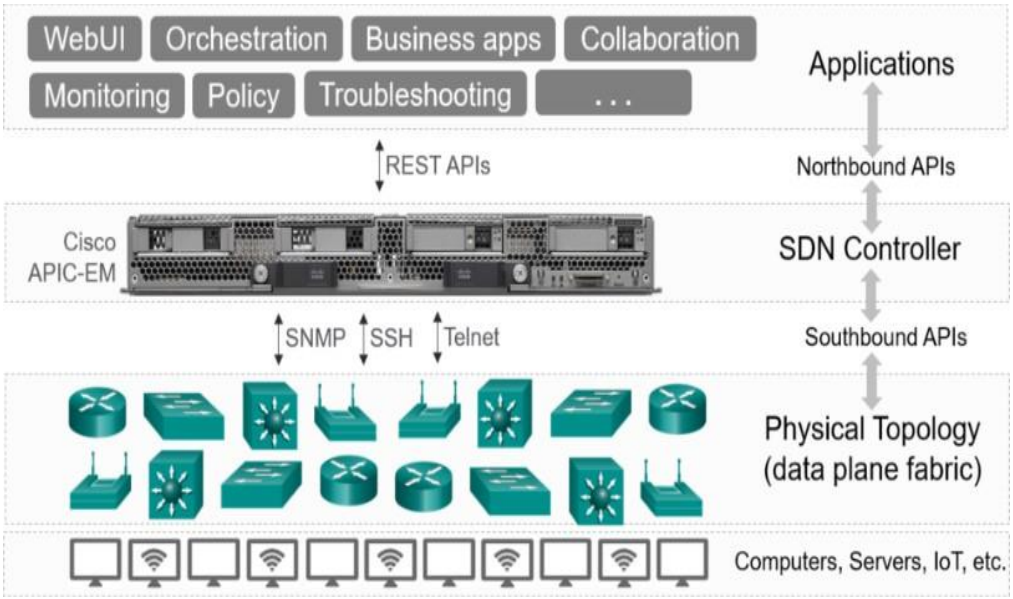


Рисунок 2.16 - контролер APCI-EM компанії Cisco System[9]

Для можливості в комутації між мережами в центральному і переферійному офісах, компанія Cisco Systems рекомендує комутатори серії 2960 з підтримкою PoE / UPoE або серій 3650/3850, що включають також функції контролера бездротового доступу. У головному офісі слід передбачити від 8 до 20 комутаторів (по 24 або 48 портів) — але потрібно пам′ятати, що комплектація комутатора розраховується згідно розробленої СКС, наявністю ЦОДа і вимог по підтримці бездротової мережі. У віддалених офісах пропонується встановити два-три комутатора зазначених серій.

Можливості підключення головного і переферійних офісів забезпечать маршрутизатори серії ISR 4000. У головному офісі рекомендується встановлювати відмовостійку пару маршрутизаторів типу Cisco ISR 4451, а у переферійних офісах (в залежності від вимог відмовостійкості і можливостей каналів глобальної мережі) — один або два маршрутизатора ISR 4431.

Для забезпечення відмовостійкості Cisco Systems рекомендує розгортати контролер APIC-EM на віртуальних машинах, причому, вони повині бути рознесені територіально.

Додатковими можливостями, компанія Cisco Systems зазначає: автоматичне виявлення і налаштування нових підключених мережевих пристроїв (для цього використовуються протоколи CDP / LLDP, а також функціонал PnP-сервера з боку APIC-EM і PnP-клієнта з боку телекомунікаційного обладнання), взаємодія з системами уніфікованих комунікацій, автоматизоване забезпечення Call Admission Control (CAC), автоматизацію мережевої безпеки (при інтеграції з зовнішніми системами).

Серед додаткових можливостей, що надаються рішенням на базі APIC-EM, представники Cisco виділили автоматичне виявлення і налаштування нових мережевих пристроїв (для цього використовуються протоколи CDP / LLDP, а також функціонал PnP-сервера з боку APIC-EM і PnP- Додатковими можливостями, компанія Cisco Systems зазначає: автоматичне виявлення і налаштування нових підключених мережевих пристроїв (для цього використовуються протоколи CDP / LLDP, а також функціонал PnP-сервера з боку APIC-EM і PnP-клієнта з боку телекомунікаційного обладнання), взаємодія з системами уніфікованих комунікацій, автоматизоване забезпечення Call Admission Control (CAC), автоматизацію мережевої безпеки (при інтеграції з зовнішніми системами).

Cisco Systems поставляє контролери APIC-EM (рис 2.17) безкоштовно з набором стандартних мережевих додатків (за нові спеціалізовані додатки буде, стягуватися додаткова платня).

Компанія Cisco Systems стверджує, що при використанні її обладнання інтеграція з традиційної мережі відбувається без особливих труднощів і навіть надаються послуги консалтингу у цьому питанні безкоштовно. Підключення до існуючої мережі рекомендується проводити через два опорних маршрутизатора Cisco ISR 4451, що слід розташовувати в головному офісі. Дивлячись на те що маршрутизатори стануть керуватися контролером APIC-EM, для взаємодії з традиційною частиною мережі використовуються стандартні механізми — протоколи канального рівня і

протоколи комутації / маршрутизації, відповідно до корпоративних стандартів .

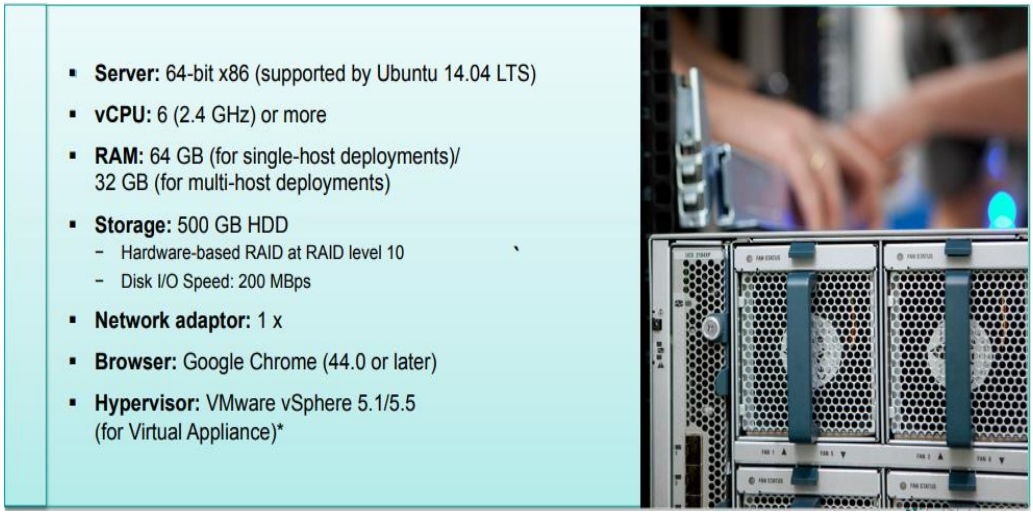


Рисунок 2.17 - Характеристики контролера APIC-EM компанії Cisco System[11]

Компанія Cisco Systems умовно поділила архітектуру роботи контролера APIC-EM на чотири рівні:

- На першому рівні працюють додатки контролера APIC-EM. Ці додатки постачаються разом з контролером. Також рівень включає в себе різні базові додатки, які є безкоштовними, та IWAN Aplication, який ліцензується окремо;
- Другий рівень – це документований REST API;
- Третій рівень – це сервіси APIC-EM. Він включає в себе базові та специфічні сервіси;
- Останій четвртий рівень – це платформа для маштабування та розширення.

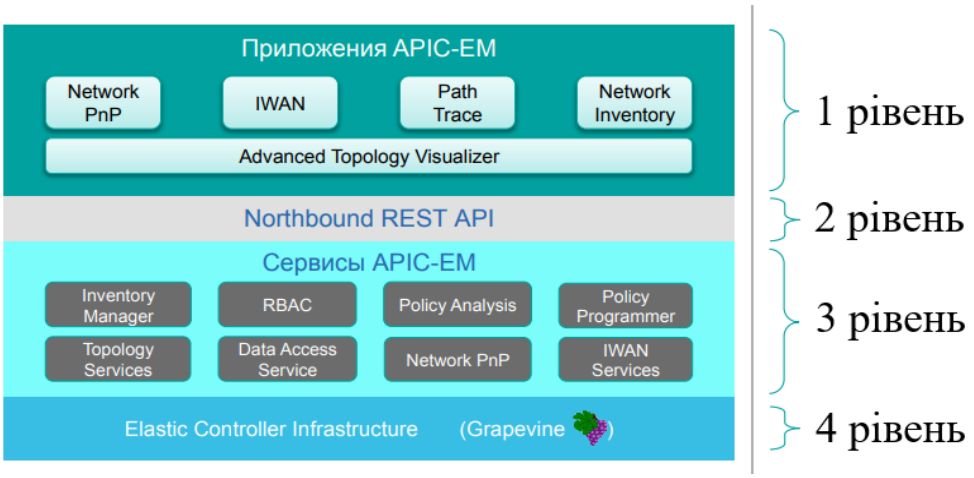


Рисунок 2.18 - Рівні архітектури від компанії Cisco System

2.3 Використання контролера APIC-EM фірми Cisco

Модуль Cisco® Application Policy Infrastructure Controller (APIC) для мережевих підприємств - це програмний контролер, який використовує управління системою, реалізує підтримку IT нового покоління. Контролер підтримує суттєву мережеву інфраструктуру, забезпечуючи захист інвестицій. Він надає програмний інтерфейс для форуму, для політик та безпеки, який пропонує доступ до сервісу в мережах комутаторів Cisco Catalyst®, маршрутизаторів з інтегрованими сервісами Cisco ISR та маршрутизаторами агрегації Cisco ASR. З підтримкою цього контролера (рис.2.19) організації можуть використати структуру власних мереж і автоматично керувати фізичною та віртуальною мережами з будь-яких комп'ютерів (Unix чи Windows) на базі архітектури x86 або з підтримкою віртуалізації.[12]

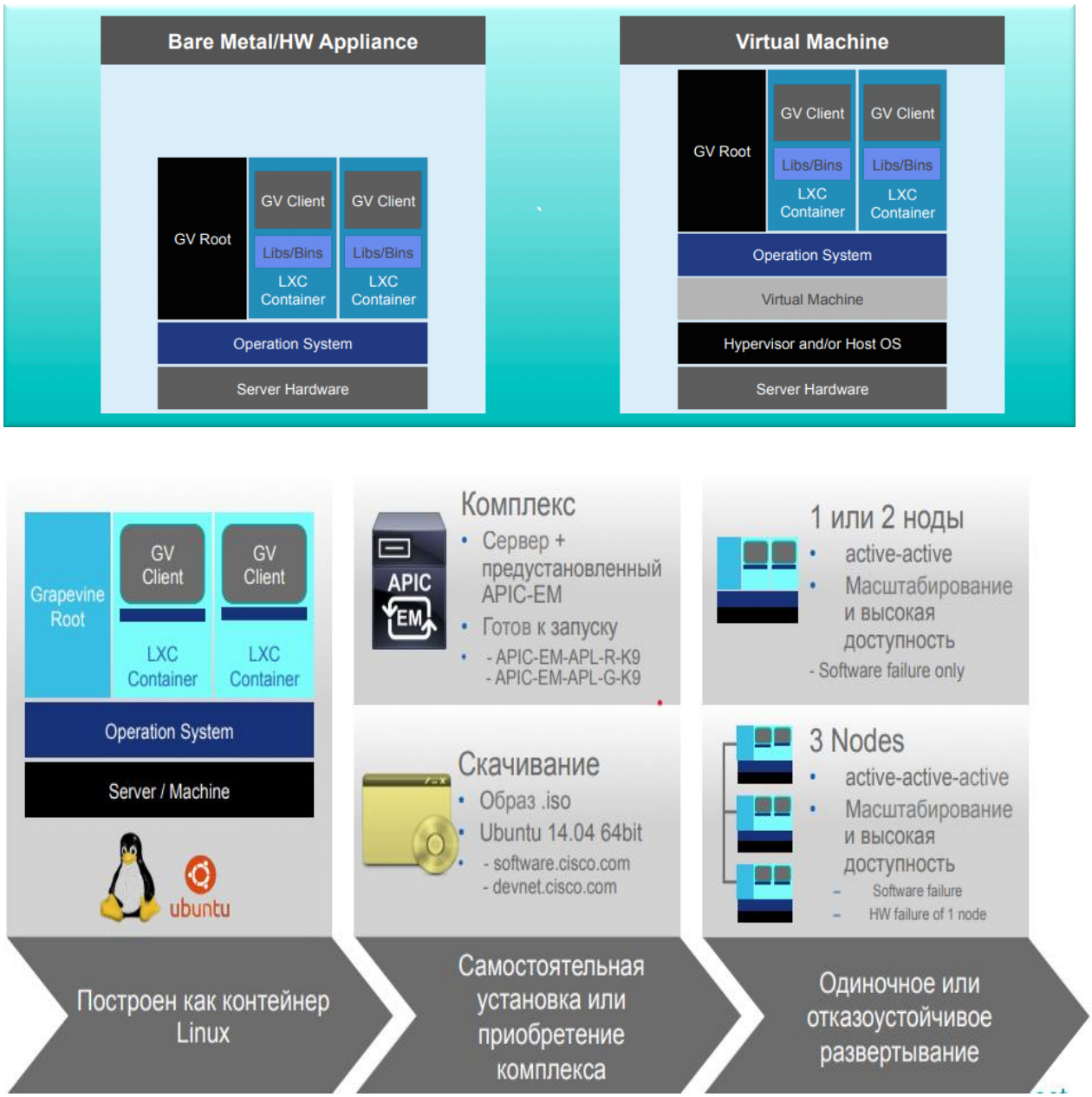


Рисунок 2.19 - Методи розгортання контролера APIC-EM компанії Cisco System

Контролер підтримує такі пристрої LAN:

- Catalyst 2960-X/XR Series Switches
- Catalyst 2960-S Series Switches
- Catalyst 2960 Series Compact Switches
- Catalyst 3560 Series Compact Switches C
- Catalyst 3850 Series Switches
- Catalyst 3750-X Series Switches
- Catalyst 3560-X Series Switches

- Catalyst 4500 Series Switches
- Catalyst 4500x Series Switches
- Catalyst 4900 Series Switches
- Catalyst 6500 Series Switches
- Catalyst 6800 Series Switches
- Cisco Nexus 5000 Series Switches
- Cisco Nexus 7000 Series Switches
- EtherSwitch Modules for Integrated Services Routers: SM-E22-16-P, SMES2-24-P, SM-D-ES2-48, SM-E
- Industrial Ethernet 2000 Series Switches
- Industrial Ethernet 3000 Series Switches

Контролер підтримує такі пристрої WAN:

- 4000 Series Integrated Services Routers
- Integrated Services Routers Generation 2
- ASR 1000 Series Aggregated Services Routers
- ASR 9000 Series Aggregated Services Routers
- Cisco Cloud Services Router 1000v

Контролер підтримує такі пристрої WLAN:

- Wireless LAN Controllers (IOS XE & AireOS)

2.3 Висновки до розділу 2

Cisco APIC-EM, на відміну від класичних концептів SDN контролерів, має низку істотних переваг. По-перше, поряд з можливістю використання REST API для програмування послуг в мережі, APIC-EM має інтуїтивно зрозумілий графічний інтерфейс управління - будь-яка функція доступна через API, також доступна і через графічний інтерфейс і навпаки. Це дозволяє ефективно використовувати APIC-EM в рамках бімодального IT, коли традиційний IT взаємодіє з інфраструктурою в основному через графічний інтерфейс, а підрозділи DevOps - через API. По-друге, APIC-EM

можна почати використовувати в мережі, що вже перебуває в експлуатації - на відміну від класичних SDN-рішень площині управління (Control Plane) не потрібно мігрувати на централізований SDN-контролер, що дозволяє одночасно зберегти автономність кожного пристрою, а з іншої сторони приховати складність інфраструктури. Це особливо важливо з урахуванням розподіленої природи корпоративної мережі, для якої класичні SDN-рішення з централізацією площині управління мають обмежені застосування. По-третє, вже зараз Cisco APIC-EM підтримує всі сучасні моделі обладнання Cisco для корпоративної мережі. Нарешті, APIC-EM відразу поставляється з набором готових додатків, покликаних автоматизувати операції, що найбільш часто зустрічаються в корпоративній мережі, - впровадження нового обладнання (додаток Network PnP), застосування Cisco CVD дизайнів і політик (додатки IWAN App та EasyQoS App), пошук несправностей і збоїв в мережі (додаток Path Tracer) та інші.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЗАЄМОДІЇ ПРОГРАМ НА БАЗІ КОНТРОЛЛЕРА APIC-EM

3.1 Опис процесу налаштування обладнання

Для демонстрації можливостей контролера APIC-EM від фірми Cisco, скористаймося безкоштовними сервісом, який надає компанія. Доступ до середовищ пісочної програми APIC-EM, розміщених на DevNet, використовуючи таку URL-адресу: [13]

Логін: *devnetuser*

Пароль: *Cisco123!*

Перше, при розгортанні контролера APIC-EM від фірми Cisco використовується домашня сторінка, яка являє собою інформативну панель, де є 7 основних клавіш.(рис.3.1)

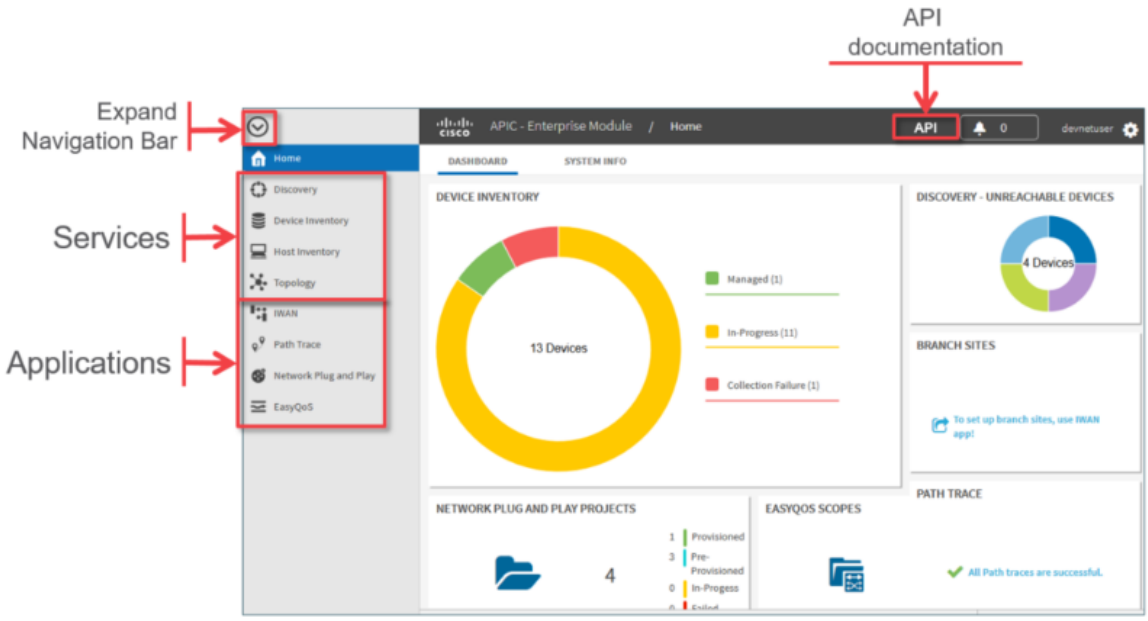


Рисунок 3.1 - Домашня сторінка контролера APIC-EM

Services - APIC-EM виявляє пристрої в мережі та розміщує їх у таблицях інвентаризації хостів та пристроїв, де можна переглянути деталі для кожного пристрою. APIC-EM також створює топологію.

Applications - Програми APIC-EM дозволяють конфігурувати мережу та перевіряти підключення до мережі. До заявок належать:

- IWAN - Спрощує розгортання WAN, надаючи інтуїтивний, заснований на політиці інтерфейс, який допомагає IT-абстрактній складності мережі та дизайну для бізнес-намірів.
- Path Trace - значно полегшує та прискорює завдання моніторингу з'єднань та усунення несправностей.
- Network Plug and Play - Забезпечує єдиний підхід до надання корпоративних мереж, що складається з маршрутизаторів, комутаторів і бездротових точок доступу Cisco, які мають досвід розгортання майже з нуля.
- Easy QoS - Надає простий спосіб класифікації та призначення пріоритету програми.

API documentation - Посилання, де можна отримати доступ до інформації про API, яка є важливою для мережевих програмістів.

Є можливість будувати графічн модель побудови мереж (рис.3.2).

Recall that Representational State Transfer (REST) - це стандартизований спосіб дозволяти системам в Інтернеті взаємодіяти між собою. Він використовує стандартні методи HTTP та транспорт без стану до даних POST або PUT та запитів до API та для отримання відповідей від API. (рис. 3.3)

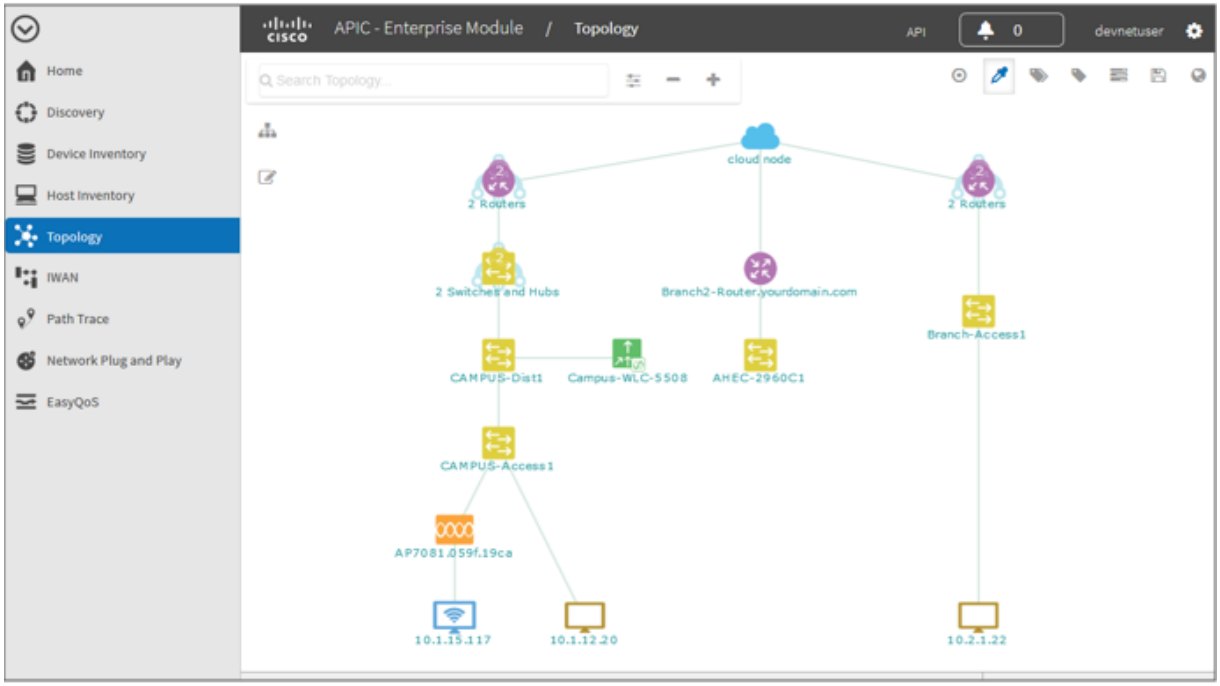


Рисунок 3.2 - Графічне уявлення мережі

API REST містять такі характеристики:

- Використовує методи протоколу HTTP та транспорт.
- Кінцеві точки API існують як серверні процеси, доступ до яких здійснюється через URL.[13]
- Веб-сторінки представляють дані та функціональність у взаємодії людина-машина, керована користувачем.
- API представляють дані та функціональність у взаємодії машина-машина, керована програмним забезпеченням.

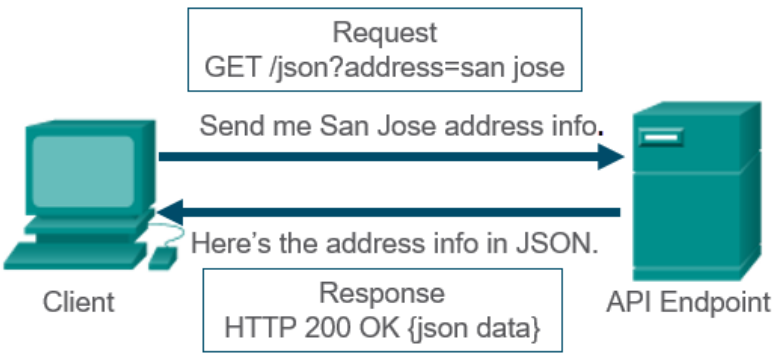
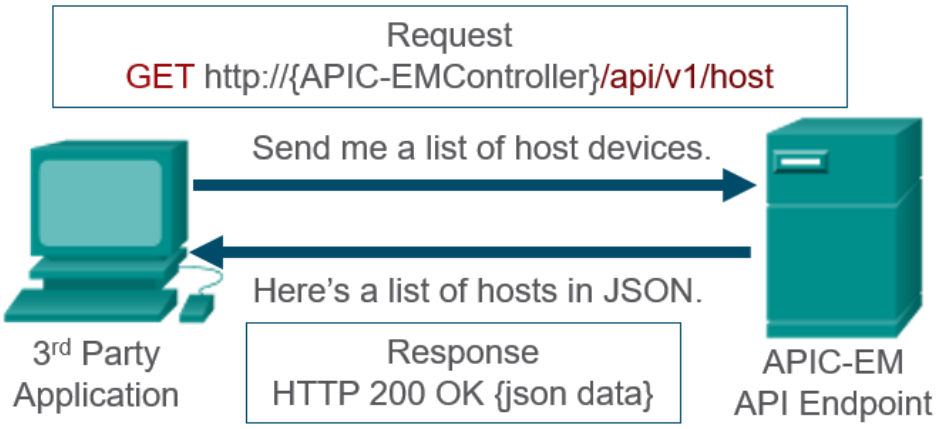


Рисунок 3.3 - Приклад запиту і відповіді від ПК до сервера

Додаток використовується для надсилання запитів (рис.3.3) до кінцевої точки API, яка приймає HTTP-запити. Кожна кінцева точка може мати різні вимоги до формату REST-запиту. Ці вимоги доступні в документації API, яка надається розробникам.

На рисунку 3.4 показана загальна URL-адреса APIC-EM на шляху до кінцевої точки, що називається хостом. Кінцева точка хоста забезпечує інвентаризацію хостів у мережі, яка приєднана до APIC-EM. Якщо запит GET правильно відформатований та автентифікований, API поверне детальну інформацію про хости у вигляді даних, відформатованих JSON.



Рисунку 3.4 - Приклад запиту від контролера APIC-EM

REST - запити вимагають наступних елементів (вимоги можуть відрізнятися залежно від API):

- Method: GET, POST, PUT, DELETE
- URL: (Як приклад) http://{APIC-EMController}/api/v1/host
- Authentication: Basic HTTP, OAuth, none, Custom
- Custom Headers: HTTP Headers(Приклад: Content-Type: application/JSON)
- Request Body: JSON або XML містять дані, необхідні для завершення запиту

Відповідь REST включає такі елементи:

- HTTP Status Code:
  - 200 OK

- 201 Created
- 401, 403 Authorization error
- 404 Resource not found
- 500 Internal Error

- Заголовки(Headers)
- Тіло(Body)
  - JSON
  - XML

Рисунок 3.5 - Приклад відповіді з API виклику, що відображається в додатку Postman

Автентифікація RESTful запиту проводиться одним із чотирьох способів:

- None: Ресурс API є загальнодоступним, і будь-хто може розмістити запит.
- Basic HTTP: Ім'я користувача та пароль передаються серверу в кодованому рядку. Цей метод є менш поширеним, ніж токени та аутентифікація OAuth.
- Token: Таємний ключ, як правило, отриманий з порталу розробників Web API.

- Open Authorization (OAuth): Відкритий стандарт для отримання маркера доступу від постачальника даних. Потім маркер передається при кожному виклику API.

APIC-EM використовує Token для управління автентифікацією. APIC-EM називає цей маркер службовим квитком. Після отримання токена (службового квитка) він використовується замість облікових даних облікових записів.

3.2 Обладнання та схема досліджень

Для провелення практичної реалізації взаємодії програм на базі котролера APIC-EM від компанії Cisco Systems створимо невеличку корпоративну мережу (рис.3.6).

Таблиця 3.1 - схема адресації корпоративної мережі

Назва пристрою	IP Address
AP7081.059f.19ca	10.1.14.3
Branch-Access1	10.2.1.17
Branch-Router1	10.2.2.1
Branch-Router2	10.2.2.2
Branch2-Router.yourdomain.com	218.1.100.100
CAMPUS-Access1	10.1.12.1
CAMPUS-Core1	10.1.7.1
CAMPUS-Core2	10.1.10.1
CAMPUS-Dist1	10.255.1.5
CAMPUS-Dist2	10.1.11.1
CAMPUS-Router1	10.1.2.1
CAMPUS-Router2	10.1.4.2
Campus-WLC-5508	10.1.14.2

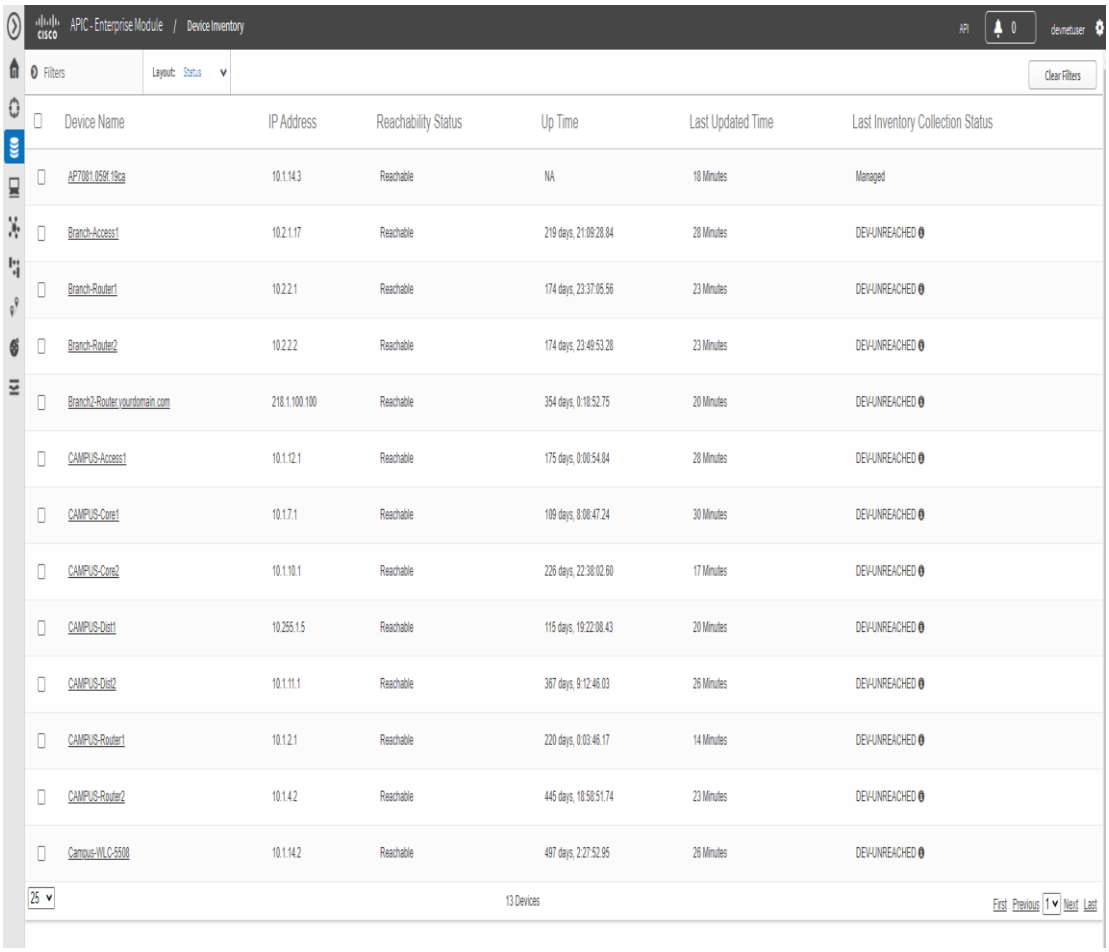


Рисунок 3.6 - Перелік пристроїв мережі

Можливо створити наявну топологія в графічному вигляді (рис.3.7).

Беремо стандартну задачу по побудові трасування (trace route), в якій потрібно буде виконати такі дії (рис.3.8):

- Відображає списки всіх хостів та мережевих пристроїв у мережі APIC-EM.
- Приймає введення користувача для джерел та пристроїв призначення для Path Trace.
- Ініціює Path Trace.
- Моніторить стан Path Trace, поки програма APIC-EM не буде завершена.
- Відображає інформацію користувачеві про завершене трасування шляху.

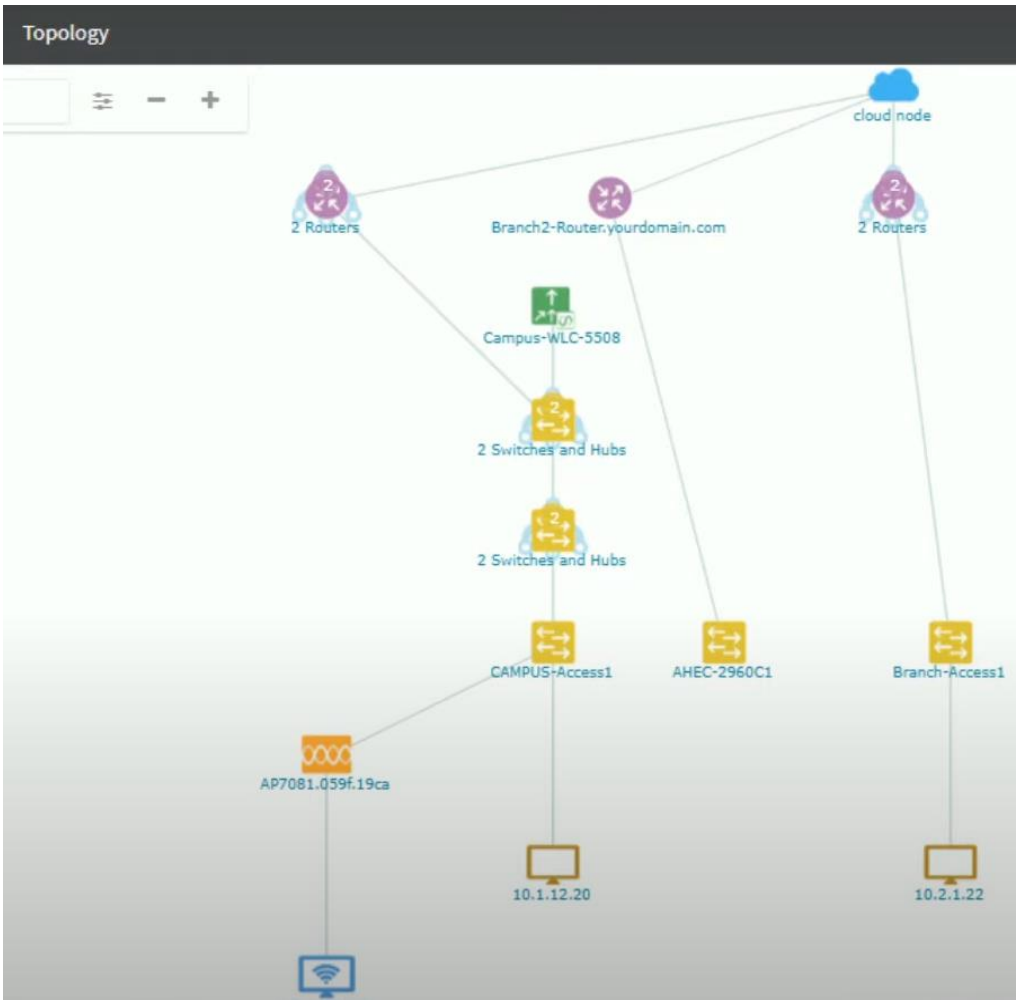


Рисунок 3.7 - Топологія корпоративної мережі

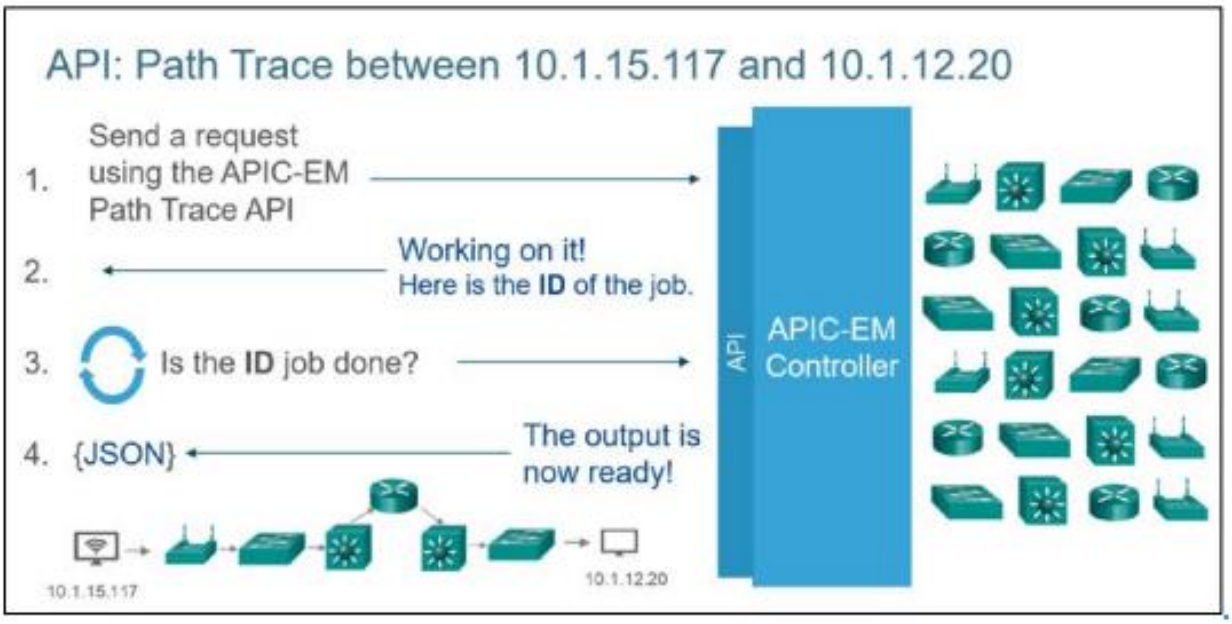


Рисунок 3.8 - Типове завдання по трасуванню

Для виконання цього завдання буде створена програма на мові програмування Python, яка взаємодіє із програмою APIC-EM Path Trace.

3.3 Інтерпретація отриманих результатів

Код для створення трасування виглядає наступним чином:

```
import requests
import json
import time

from my_api_em_functions import *
from tabulate import *

requests.packages.urllib3.disable_warnings()

api_url = "https://devnetsbx-netacad-apicem-1.cisco.com/api/v1/flow-analysis"


ticket = get_ticket()

headers = {
    "content-type": "application/json",
    "X-Auth-Token": ticket
    }

print("\n\n")
print("List of hosts on the networks: ")
print_hosts()
print("\n\n")
print("List of network devices of the network: ")
print_device()
print("\n\n")

while True:
    s_ip = input("Enter the source host IP address for the path trace: ")
    d_ip = input("Enter the destination host IP address for the path trace:")

    if s_ip != "" and d_ip != "":
        path_data = {
            "sourceIP": s_ip,
```

```
            "destIP": d_ip
            }
        print("Source IP address is: ", path_data["sourceIP"])
        print("Destination IP address is: ", path_data["destIP"])
        break
    else:
        print("\n You must enter correct IP address! \n Enter CTRL + C to exit! \n")
        continue

path = json.dumps(path_data)
resp = requests.post(api_url, path, headers = headers, verify=False)
resp_json = resp.json()
flowAnalysisId = resp_json["response"]["flowAnalysisId"]
print("Flow Analysis ID: ", flowAnalysisId)

# s6
check_url = api_url + "/" + flowAnalysisId

status = ""
checks = 1
while status != "COMPLETED":
    r = requests.get(check_url, headers=headers, verify=False)
    response_json = r.json()

    status = response_json["response"]["request"]["status"]

    print("REQUEST STATUS: ", status)

    time.sleep(1)
    if checks == 15:
        raise Exception ("Number of status crck exeeds limit. Possible problem with
Path Trace.! ")
    elif status == "FAILED":
        raise Exception("Problem with Path Trace - FAILED! ")
    checks += 1

path_source = response_json["response"]["request"]["sourceIP"]

path_dest = response_json["response"]["request"]["destIP"]

networkElementsInfo = response_json["response"]["networkElementsInfo"]
```

```
all_devices = []
device_no = 1

for networkElement in networkElementsInfo:
    if "name" not in networkElement:
        name = "Unnamed Host"
        ip = networkElement["ip"]
        devtype = networkElement["type"]
        egressInterfaceName = "UNKNOWN"
        ingressInterfaceName = "UNKNOWN"
    else:
        name = networkElement["name"]
        devtype = networkElement["type"]
        ip = networkElement["ip"]
        if "egressInterface" in networkElement:
            egressInterfaceName =
networkElement["egressInterface"]["physicalInterface"]["name"]
        else:
            egressInterfaceName = "UNKNOWN"
        if "ingressInterface" in networkElement:
            ingressInterfaceName =
networkElement["ingressInterface"]["physicalInterface"]["name"]
        else:
            ingressInterfaceName = "UNKNOWN"
    device = [
        device_no,
        name,
        devtype,
        ip,
        ingressInterfaceName,
        egressInterfaceName
        ]

    all_devices.append(device)
    device_no += 1

print("Path Trace: \n Sourse: ", path_source,"\n Destination: ",path_dest)

print("List of device on path: ")

table_header = [
    "Item",
```

```
    "Name",
    "Type",
    "IP",
    "Ingress Int",
    "Egress Int"
    ]

print(tabulate (all_devices, table_header) )
```

Розглянемо кожну частину коду окремо.

- З самого початку потрібно імпортувати бібліотеки (рис.3.9).

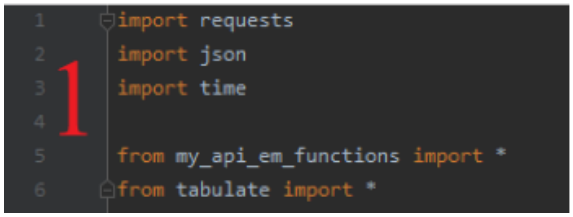


Рисунок 3.9 - Початок коду

- З посилання на контролер виймаємо всю необхідну інформацію.

- `requests.packages.urllib3.disable_warnings()`

```
    api_url = "https://devnetsbx-netacad-apicem-1.cisco.com/api/v1/flow-analysis"

    ticket = get_ticket()

    headers = {
        "content-type": "application/json",
        "X-Auth-Token": ticket
        }

    print("\n\n")
    print("List of hosts on the networks: ")
    print_hosts()
    print("\n\n")
    print("List of network devices of the network: ")
    print_device()
    print("\n\n")
```

- Запитуємо в користувача IP адреси, які портібно ввести.

```
while True:
    s_ip = input("Enter the source host IP address for the path trace: ")
    d_ip = input("Enter the destination host IP address for the path trace:")

    if s_ip != "" and d_ip != "":
        path_data = {
            "sourceIP": s_ip,
            "destIP": d_ip
            }
        print("Source IP address is: ", path_data["sourceIP"])
        print("Destination IP address is: ", path_data["destIP"])
        break
    else:
        print("\n You must enter correct IP address! \n Enter CTRL + C to
exit! \n")
        continue

path = json.dumps(path_data)
resp = requests.post(api_url, path, headers = headers, verify=False)
resp_json = resp.json()
flowAnalysisId = resp_json["response"]["flowAnalysisId"]
print("Flow Analysis ID: ", flowAnalysisId)
```

- Перевіряємо статус нашого запиту.

```
check_url = api_url + "/" + flowAnalysisId

status = ""
checks = 1
while status != "COMPLETED":
    r = requests.get(check_url, headers=headers, verify=False)
    response_json = r.json()

    status = response_json["response"]["request"]["status"]

    print("REQUEST STATUS: ", status)

    time.sleep(1)
    if checks == 15:
        raise Exception ("Number of status crck exeeds limit. Possible problem
with Path Trace.! ")
    elif status == "FAILED":
```

```
            raise Exception("Problem with Path Trace - FAILED! ")
        checks += 1

    path_source = response_json["response"]["request"]["sourceIP"]

    path_dest = response_json["response"]["request"]["destIP"]


    networkElementsInfo = response_json["response"]["networkElementsInfo"]

    all_devices = []
    device_no = 1
```

- Робимо обхід по всіх пристроях і вписуємо їх в таблицю.

```
- for networkElement in networkElementsInfo:
        if "name" not in networkElement:
            name = "Unnamed Host"
            ip = networkElement["ip"]
            devtype = networkElement["type"]
            egressInterfaceName = "UNKNOWN"
            ingressInterfaceName = "UNKNOWN"
        else:
            name = networkElement["name"]
            devtype = networkElement["type"]
            ip = networkElement["ip"]
            if "egressInterface" in networkElement:
                egressInterfaceName =
    networkElement["egressInterface"]["physicalInterface"]["name"]
            else:
                egressInterfaceName = "UNKNOWN"
            if "ingressInterface" in networkElement:
                ingressInterfaceName =
    networkElement["ingressInterface"]["physicalInterface"]["name"]
            else:
                ingressInterfaceName = "UNKNOWN"
        device = [
            device_no,
            name,
            devtype,
            ip,
            ingressInterfaceName,
            egressInterfaceName
```

```
        ]

    all_devices.append(device)
    device_no += 1

print("Path Trace: \n Sourse: ", path_source,"\n Destination: ",path_dest)

print("List of device on path: ")

table_header = [
    "Item",
    "Name",
    "Type",
    "IP",
    "Ingress Int",
    "Egress Int"
    ]

print(tabulate (all_devices, table_header) )
```

Результатом всіх маніпуляцій з кодом вийде наступне. Спочатку програма запитає в користувача IP адреси, за якими потрібно провести трасування. У прикладі – це від 10.1.15.117 до 10.1.12.20 (рис.3.10 і 3.11).

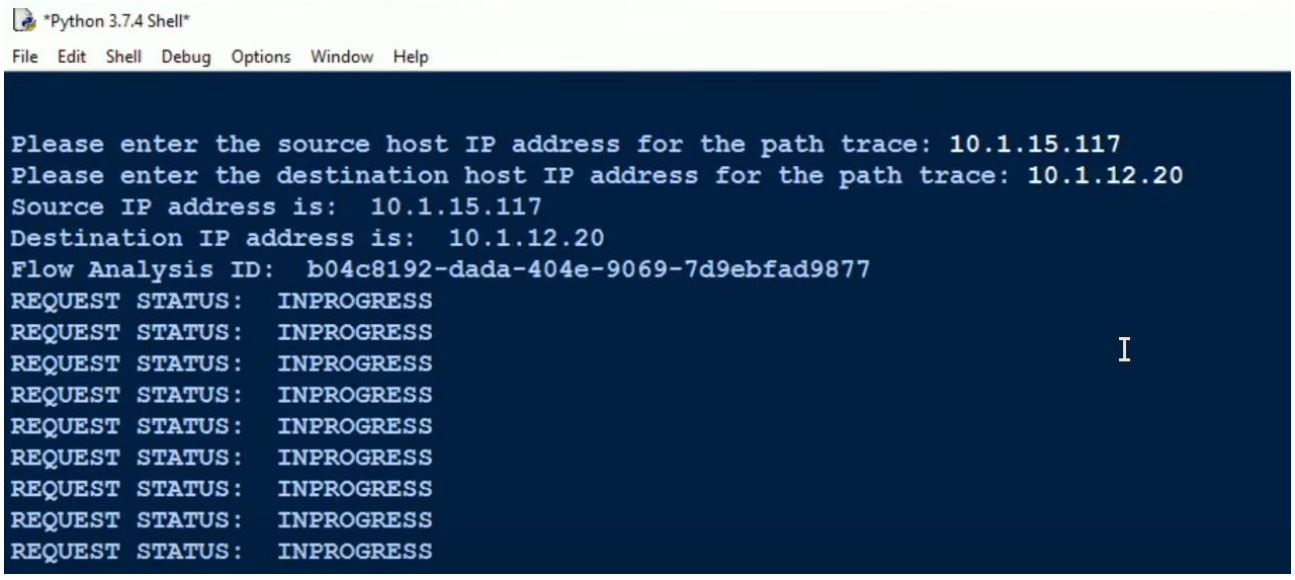


Рисунок 3.10 - Приклад програми запиту

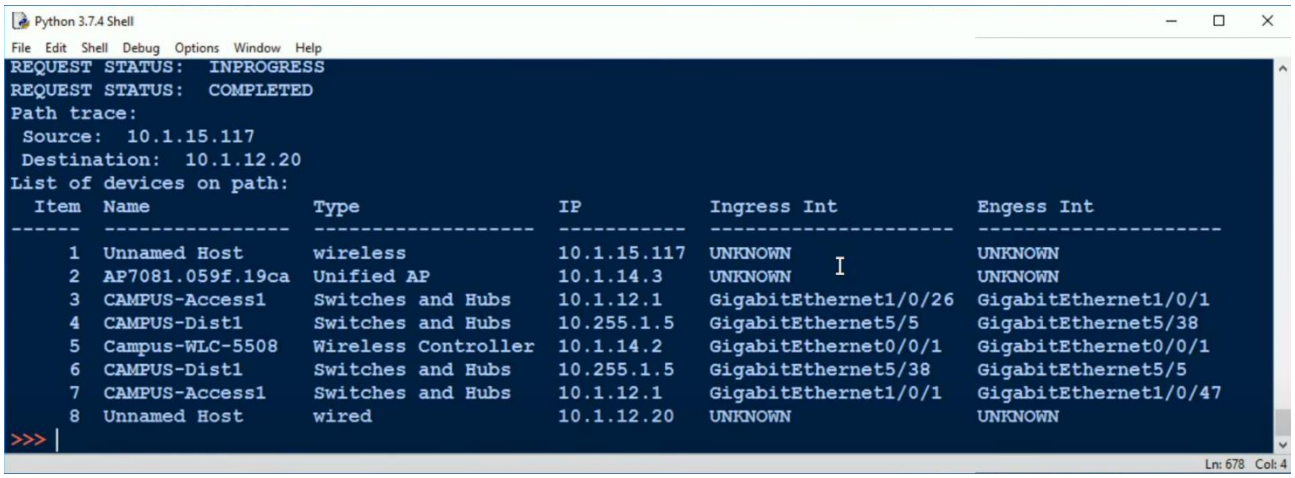


Рисунок 3.11 - Результат програми

З рисунка 3.11 бачимо, що програма пройшла по всій топології та створила таблицю, в якій вказала назву пристрою, тип, IP адресу, порти на вхід та вихід.

3.4 Висновки до розділу 3

Висновком розділу зазначимо те, що контролер APIC-EM – це універсальний механізм, який легко інтегрується в мережу, за допомогою нього можна швидко налаштовувати корпоративні мережі. При використанні APIC-EM перевагою є те, що більшість складних операцій можна виконати лише один раз, а потім просто запускати вже існуючі шаблоні сценарії. Контролер самостійно виявляє підключені до мережі пристрої, та зчитує технічну інформацію на них. Використовуючи APIC-EM від фірми Cisco, компанія інвестує не тільки в короткострокові переваги, а й робить заділ на майбутнє, бо технології SDN тільки починають набирати свої оберти, і щоб не відстати від часу, потрібно вже зараз замислюватися про впровадження механізмів, які в майбутньому дадуть змогу швидко інтегрувати корпоративні мережі в нові і більш сучасні.

ВИСНОВКИ

Підводячи підсумок, зазначимо, що SDN — це революційна архітектура IT для будь-якого сучасного підприємства. Контролер APIC-EM дозволяє замовникам радикально знизити витрати при впровадженні та експлуатації IT-інфраструктури та послуг, прискорити процес інновацій в IT, при цьому забезпечити захист існуючих і майбутніх інвестицій і, що важливо, IT стати частиною основного бізнесу і забезпечувати його зростання та передбачуваність.

Основна особливість SDN-мереж полягає у відділенні площини управління від площини даних. Як площина управління виступає SDN-контролер, на який переноситься все навантаження по управлінню потоком з маршрутизаторів і комутаторів, на відміну від традиційних мереж. Площина управління містить механізми переадресації маршруту рівня 2 та 3. Контролер SDN управляє станом пересилання комутаторів в SDN. Це управління здійснюється за допомогою API, що дозволяє контролеру задовольняти найрізноманітніші вимоги додатка без зміни будь-яких аспектів нижчого рівня мережі. Також завдяки розподілу площин управління і даних SDN дозволяє програмам працювати з одним абстрактним мережевим пристроєм, не піклуючись про деталі роботи пристрою.

Використання контролерів дає змогу економити на людських ресурсах і не витрачати час і гроші на підготовку великої кількості персоналу. Контролер APIC-EM відразу поставляється з набором готових додатків, покликаних автоматизувати операції, що найбільш часто зустрічаються в корпоративній мережі, - впровадження нового обладнання (додаток Network PnP), застосування Cisco CVD дизайнів і політик (додатки IWAN App та EasyQoS App), пошук несправностей і збоїв в мережі (додаток Path Tracer) та інші. Також однією з найбільших переваг є мобільність перенаправлення трафіку, за потреби можливе балансування і швидка зміна трафіку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Глобальне соціальне креативне агентство. – Режим доступу до ресурсу: https://wearesocial.com/
2. Сетевые технологии SDN – Software Defined Networking [Електронний ресурс] // 4 березня 2015. – Режим доступу до ресурсу: https://habr.com/ru/company/muk/blog/251959/
3. Гніденко М.П. Побудова SDN мереж // Вишнівськтй В.В., Ільїн О.О. – КИЇВ, 2019, - 190 стр. – Режим доступу до ресурсу: http://www.dut.edu.ua/uploads/l_1710_34882811.pdf
4. SDN: от концепции к решениям [Електронний ресурс] // 2015. – Режим доступу до ресурсу: https://www.osp.ru/lan/2015/09/13046914/
5. Google trends https://trends.google.com/trends/?geo=US
6. Тренди Google виборі мови програмування. https://trends.google.com/trends/explore?q=%2Fm%2F05z1_,%2Fm%2F06ff5,%2Fm%2F07sbkfb
7. Головна сторінка Python https://www.python.org/doc/
8. Изучаем Python/ Четвертое издание/ Mark Lutz/ 2011г.
9. Контролер APCI-EM компанії Cisco System. Електронний ресурс – Режим доступу до ресурсу: https://www.cisco.com/c/en/us/products/cloud-systems-management/application-policy-infrastructure-controller-enterprise-module/index.html
10. Характеристики контролера APIC-EM компанії Cisco System. Електронний ресурс – Режим доступу до ресурсу: https://www.cisco.com/c/dam/assets/global/RU/events/cisco-connect/presentation/ural/18/1530.pdf
11. СРАВНИТЕЛЬНЫЙ АНАЛИЗ SDN-КОНТРОЛЛЕРОВ [Електронний ресурс] // 7. – 2017. – Режим доступу до ресурсу: https://cyberleninka.ru/article/n/sravnitelnyy-analiz-sdn-kontrollerov

12. Спеціалізований веб-сайт для розробників ПО від компанії Cisco System https://developer.cisco.com/
13. Доступ до ресурсів контролера APIC-EM компанії Cisco System. Електронний ресурс – Режим доступу до ресурсу: https://sandboxapicem.cisco.com/